Adaptive Sparse Coding and Dictionary Selection

Mehrdad Yaghoobi Vaighan

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
January 8, 2010

Abstract

The sparse coding is approximation/representation of signals with the minimum number of coefficients using an overcomplete set of elementary functions. This kind of approximations/ representations has found numerous applications in source separation, denoising, coding and compressed sensing. The adaptation of the sparse approximation framework to the coding problem of signals is investigated in this thesis. Open problems are the selection of appropriate models and their orders, coefficient quantization and sparse approximation method. Some of these questions are addressed in this thesis and novel methods developed. Because almost all recent communication and storage systems are digital, an easy method to compute quantized sparse approximations is introduced in the first part.

The model selection problem is investigated next. The linear model can be adapted to better fit a given signal class. It can also be designed based on some a priori information about the model. Two novel dictionary selection methods are separately presented in the second part of the thesis. The proposed model adaption algorithm, called Dictionary Learning with the Majorization Method (DLMM), is much more general than current methods. This generality allowes it to be used with different constraints on the model. Particularly, two important cases have been considered in this thesis for the first time, Parsimonious Dictionary Learning (PDL) and Compressible Dictionary Learning (CDL). When the generative model order is not given, PDL not only adapts the dictionary to the given class of signals, but also reduces the model order redundancies. When a fast dictionary is needed, the CDL framework helps us to find a dictionary which is adapted to the given signal class without increasing the computation cost so much.

Sometimes a priori information about the linear generative model is given in format of a parametric function. Parametric Dictionary Design (PDD) generates a suitable dictionary for sparse coding using the parametric function. Basically PDD finds a parametric dictionary with a minimal dictionary coherence, which has been shown to be suitable for sparse approximation and exact sparse recovery.

Theoretical analyzes are accompanied by experiments to validate the analyzes. This research was primarily used for audio applications, as audio can be shown to have sparse structures. Therefore, most of the experiments are done using audio signals.

to my parents,

Haedeh and Houshang

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Department of Electronics and Electrical Engineering at The University of Edinburgh.

The research on different dictionary learning methods using the majorization minimization method, detailed in Chapter 5, was done in collaboration with Dr. Thomas Blumensath, now at the University of Southampton. The research on parametric dictionary design, detailed in Chapter 8, was started when the author had a short visit to the laboratory of Prof. Laurent Daudet, UPMC Paris 6. The work performed here was a collaboration with him and his research assistant Dr. Emmanuel Ravelli, now at the University of California at Santa Barbara (UCSB).

Mehrdad Yaghoobi Vaighan

Acknowledgments

This thesis has been prepared with the help of many people, which is impossible to acknowledge all of them here. I would like to begin by thanking my supervisor Prof. Mike E. Davies, who has been more than a supervisor to me. Thanks for your support, patience and open mind. I also want to thank Dr. Thomas Blumensath for our long discussions and for describing my mistakes patiently.

I started my PhD study at Queen Mary, University of London (QMUL) before moving to Edinburgh. I therefore acknowledge the support of Electrical Engineering at QMUL for waiving part of the tuition fee and people at the Center for Digital Music, especially Prof. Mark Plumbley for accepting to be my second supervisor at QMUL and commenting on my reseach during our project meetings. I would also like to thank Prof. Remi Gribonval from INRIA, France, for introducing me to MED as a possible PhD candidate.

As part of this research was started in France, I would like to acknowledge the hospitality and support of Prof. Laurent Daudet and Dr. Emmanuel Ravelli at UPMC, Paris 6.

This thesis has been prepared as the final step of a PhD program which was fully supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant D000246/1. I would like to thank the EPSRC for their financial support and travel grants, which allowed me to present the papers at several conferences.

This thesis could not be prepared as comprehensive as it is, without the support and encouragement of my parents, Dr. Haedeh Arvand and Houshang Yaghoobi Vaighan. Thanks for all you have done for me.

I also want to thank all the IDCom members at the University of Edinburgh. Special thanks to Dr. James R. Hopgood for being my second supervisor, Nicola Ferguson for continuous help and all the *Espresso Club* members, alphabetically ordered, David, George, Graham, Ioannis, Mohammad, Renato, Steve, Xiaoyan (Alice) and Xionghu for the *great* discussions at the coffee time, and Sharad for his support.

Contents

Declaration of originality v
Acknowledgments vi
Contents vii
List of figures x
Acronyms and abbreviations xii
Nomenclature xiv

1 Introduction **1**
1.1 Background 1
1.1.1 Sparse Coding 1
1.1.2 Dictionary Selection 3
1.1.3 Quantization 5
1.2 Contributions 6
1.3 Organization 8

I Sparse Coding **10**

2 Sparse Coding Formulations **11**
2.1 Introduction 11
2.2 Sparse Coding Formulations 11
2.3 Sparse Matrix Coding 16
2.4 Summary 18

3 Sparse Coding Algorithms **19**
3.1 Introduction 19
3.2 Sparse Approximation Methods 20
3.3 Greedy Methods 21
3.3.1 Matching Pursuit 22
3.3.2 Orthogonal Matching Pursuit 23
3.3.3 Gradient Pursuit 23
3.4 Relaxed Sparse Approximation Methods 24
3.4.1 Majorization Minimization Method 25
3.4.2 Iterative Thresholding 26
3.4.3 Iterative Reweighting 31
3.4.4 Other Sparse Approximation Methods 33
3.5 Quantized Sparse Approximations 34
3.5.1 Quantized Sparse Approximation with the Majorization Method 35
3.6 Simulations 37
3.7 Summary 39

II Dictionary Selection **41**

4 Dictionary Learning Formulation and State of the Art Algorithms **43**
4.1 Introduction . . . 43
4.2 Dictionary learning formulation . . . 44
4.3 Dictionary learning for sparse approximations . . . 46
4.3.1 Dictionary Learning using a Maximum Likelihood Estimator . . . 47
4.3.2 Dictionary Learning using a Maximum A Posteriori Estimator . . . 50
4.3.3 Method of Optimal Directions (MOD) . . . 53
4.3.4 K-SVD Dictionary Learning . . . 54
4.3.5 Other Dictionary Learning Methods . . . 57
4.4 Structured Dictionary Learning Methods . . . 60
4.4.1 Shift Invariant Dictionary Learning . . . 61
4.4.2 Multiscale Dictionary Learning . . . 67
4.4.3 Unions of Orthonormal Bases Dictionary Learning . . . 68
4.4.4 Other Structured Dictionary Learning Methods . . . 69
4.5 Summary . . . 73

5 Dictionary Learning with the Majorization Minimization Method **75**
5.1 Introduction . . . 75
5.2 Dictionary Learning using Majorization Minimization . . . 76
5.2.1 Matrix Valued Sparse Approximation . . . 77
5.2.2 Dictionary Update . . . 78
5.2.3 Generalized block relaxation method for dictionary learning . . . 83
5.3 Simulations . . . 84
5.3.1 Synthetic Data . . . 84
5.3.2 Dictionary Learning for Sparse Audio Coding . . . 89
5.4 Summary . . . 91

6 Parsimonious Dictionary Learning (PDL) **95**
6.1 Introduction . . . 95
6.2 Parsimonious Dictionary Learning Formulation . . . 96
6.3 PDL with the Majorization Minimization Method . . . 97
6.3.1 PDL: Dictionary Update Step . . . 97
6.4 Simulation . . . 99
6.4.1 Synthetic Data . . . 100
6.4.2 Parsimonious Dictionary Learning for Sparse Audio Coding . . . 101
6.5 Conclusions . . . 103

7 Compressible Dictionary Learning (CDL) **105**
7.1 Introduction . . . 105
7.2 Compressible Dictionary . . . 105
7.3 Problem Formulation . . . 108
7.4 CDL Algorithm . . . 109
7.4.1 Derivation of the Majorizing Functions for CDL . . . 109
7.5 Simulations . . . 111
7.6 Summary . . . 113

8 Parametric Dictionary Design (PDD) 115
8.1 Introduction . . . 115
8.2 Dictionary Design for Sparse Coding . . . 115
8.3 PDD: Formulation . . . 118
8.4 PDD: A Practical Algorithm . . . 120
8.4.1 Projection onto Λ^N: . . . 121
8.4.2 Parameter update: . . . 122
8.5 Case study . . . 124
8.5.1 Gammatone parametric dictionary . . . 124
8.5.2 Simulations results . . . 125
8.6 Summary . . . 131

9 Conclusion and Future Work 135
9.1 Overview . . . 135
9.2 Conclusion and Future Work . . . 136

A Matrix Form of the Majorizing Function 141

B Convergence Study of the Dictionary Learning with the Majorization Minimization Method 143
B.1 Generalized block relaxed iterative mappings and their convergence . . . 144
B.2 Convergence study of the generalized block relaxed dictionary learning . . . 147

C Derivation of the Gammatone Dictionary Gradient 149

D Convergence Study of the Parametric Dictionary Design 151

References 154

List of figures

3.1 9 level on-center QShrinker 36
3.2 Input audio signal 37
3.3 For two different numbers of iterations (20 and 100) output SNR's are shown in four different cases (IT (+), QIT (x), quantized QIT (*), quantized IT (o)) . . 38
3.4 Operating R-D curves for QIT (upper) and IT (lower) 39

5.1 A comparison of the dictionary recovery success rates using different dictionary learning methods under a column-norm constraint. 84
5.2 A comparison of the computation costs of the dictionary learning methods under a column-norm constraint. 85
5.3 A comparison of the dictionary recovery success rates using MM and MAP dictionary learning methods under a Frobenius norm constraint: 1: Desired dictionary had fixed Frobenius-norm. 2: Desired dictionary had fixed column-norms. 86
5.4 A comparison of the computation costs of the dictionary learning methods under a Frobenius norm constraint. 87
5.5 l_0 cost functions of the constrained Frobenius and column -norms dictionary learning algorithms respectively on top and bottom plots. 88
5.6 ℓ_1 cost functions for two different Lagrangian multipliers (λ) .005 (top) and .001 (bottom). 89
5.7 A selection of learned atoms in time (left) and frequency (middle) domain. Their norms are shown in the right panel. 90
5.8 Number of appearances of the learned atoms in the representations of the training samples (of size 8192). 92
5.9 Estimated Rate-Distortion for the audio coding example using the learned dictionary, the shrunk 2 times overcomplete DCT dictionary and the DCT. 93

6.1 Exact recovery with the constrained column-norm. 99
6.2 Exact recovery with the bounded Frobenius column-norm. 100
6.3 Number of appearances in the representations of the training blocks (of size 8192). 101
6.4 Estimated Rate-Distortion for the audio coding. 102

7.1 The atom generation in the CDL framework: (a) i^{th} column of $\mathbf{\Psi}$, ψ_i, (b) The atoms ϕ_k which are related to the non-zero values of selected ψ_i, $\{\phi_k : \psi_i(k) \neq 0\}$, (c) $\phi_k \psi_i(k) : \psi_i(k) \neq 0$, (d) The i^{th} atom of $\mathbf{D} = \mathbf{\Phi\Psi}$. 113
7.2 The sparsity $\mathcal{J}_1(.)$ vs. representation error plots of 4096 evaluation signals. . . 114

8.1 Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method. 119
8.2 The chain rule (8.11) in the tensor form. 122
8.3 The objective functions for different $\{\alpha_k\}_{\forall k, \alpha_k = \alpha}$, for a constant α. 126

8.4 Eigen values plot of the dictionary. 128
8.5 The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries. 129
8.6 Exact support recovery of the sparse signals. 130
8.7 The residual error using matching pursuit for sparse approximation of the audio signal. 131
8.8 Wigner-Ville contour plots of the original Gammatone atoms. The WV contour of each atom is calculated at 0.7 times it peak. 133
8.9 Wigner-Ville contour plots of the learned Gammatone atoms. The contours are calculated similar to Fig. 8.8 134

Acronyms and abbreviations

BODL	Block-Overlapped Dictionary Learning
BP	Basis Pursuit
BPDN	Basis Pursuit DeNoising
BSS	Blind Source Separation
CDL	Compressible Dictionary Learning
CS	Compressed Sensing
DCT	Descrete Cosine Transform
DLMM	Dictionary Learning with the Majorization Method
ECG	Electrocardiography
ERC	Exact Recovery Condition
ETF	Equiangular Tight Frame
FOCUSS	FOCal Underdetermined System Solution
GP	Gradient Pursuit
GPCA	Generalized PCA
GPSR	Gradient Projection for Sparse Representations
ICA	Independent Component Analysis
IHT	Iterative Hard Thresholding
IRLS	Iterative Least Square
IT	Iterative Thresholding
K.K.T	KarushKuhnTucker
LAR	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
MAP	Maximum A Posteriori
MAP-DL	MAP based Dictionary Learning
MCLT	Modulated Complex Lapped Transform
MCMC	Monte Carlo Markov Chain
MDCT	Modified DCT
ML	Maximum Likelihood
MM	Majorization Method

MM-DL	Majorization Minimization based Dictionary Learning
MOD	Method of Optimal Directions
MP	Matching Pursuit
NP	Nondeterministic Polynomial time
OMP	Orthogonal Matching Pursuit
OOMP	Optimizes Orthogonal Matching Pursuit
PCA	Principal Component Analysis
PDD	Parametric Dictionary Design
pdf	probability density function
PDL	Parsimonious Dictionary Learning
POCS	Projections Onto Convex Sets
QIT	Quantized Iterative Thresholding
QVSA	Quantized Value Sparse Approximation
RD	Rate Distortion
SAR	Synthetic Aperture Radar
SCA	Sparse Component Analysis
SNR	Signal to Noise Ratio
SQP	Sequential Quadratic Programming
StOMP	Stagewise OMP
SV	Singular Value
SVD	Singular Value Decomposition
VQ	Vector Quantization
WV	Wigner-Ville

Nomenclature

$\cdot, (\cdot)$	operand, which can be vector, matrix or tensor
$(\cdot)^*$	the optimal argument found by minimizing an objective
$*$	convolution
$\#$	cardinality of a set
$(\cdot)^\dagger$	pseudoinverse
$(\cdot)^T$	transpose
$\perp$	orthogonality between two vectors or two subspaces
$\propto$	equality up to a constant
$\approx$	approximately equal
$(\cdot)^{[n]}$	the operand at the nth iteration
$\lvert \cdot \rvert$	absolute value
$\mathrm{tr}\{\cdot\}$	trace of a matrix
$\mathbf{x}_i, \{\cdot\}_i$	the ith element of a vector ($\mathbf{x}$)
$\mathbf{X}_i$	the ith column of the matrix $\mathbf{X}$
$\mathbf{X}^{(i)}, \{\cdot\}^{(i)}$	the ith row of the matrix $\mathbf{X}$
$\mathbf{D}_\Lambda$	restricted dictionary to the index set Λ
$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$	the set of all $\mathbf{x}_i$'s for all $i \in \mathcal{I}$
$[\mathbf{x}_i]_{i\in\mathcal{I}}$	the matrix generated by putting all $\mathbf{x}_i$'s as columns
$\mathcal{A} \setminus \mathcal{B}$	$\mathcal{A} - \mathcal{A} \cap \mathcal{B}$
$\frac{\partial}{\partial \mathbf{x}}$	partial derivative with respect to $\mathbf{x}$
$\nabla_{\mathbf{x}}$	gradient with respect to $\mathbf{x}$
$df(\mathbf{X})^{\rightarrow \mathbf{Y}}$	directional derivative of f in direction $\mathbf{Y}$
$\mathcal{P}_{\mathcal{A}}$	orthogonal projection onto the set $\mathcal{A}$
$\mathrm{diag}(\mathbf{x})$	diagonal matrix with the elements of $\mathbf{x}$ on the main diagonal
$f(x)\big\rvert_{x_0}$	$f(x_0)$
$\mathcal{S}_\lambda(\cdot)$	soft thresholding operator
$\mathcal{H}_\lambda(\cdot)$	hard thresholding operator
$\mathcal{O}(\cdot)$	in order of the operand

$\mathbb{R}_0^+$	$\{x \in \mathbb{R}, x \geq 0\}$
$\text{class}\, C^1$	class of continuously differentiable functions
$\text{epi}(f(x), x_0)$	epigraph of the function $f(x)$ at the level x_0
c, C	constant
$\mathbf{x}$	coefficient vector
$\mathbf{y}$	signal
$\mathbf{X}$	coefficient matrix
$\mathbf{Y}$	signal matrix
$\mathscr{Y}$	$\{\mathbf{y}_i\}_{i \in \mathcal{I}}$ training samples
$\mathbf{D}$	Dictionary
$\mathcal{D}$	dictionary admissible set
μ	coherence of a dictionary
$\lvert \cdot \rvert_\epsilon$	$\sqrt{(\cdot)^2 + \epsilon}$
$\lVert \cdot \rVert_p, \ell_p : \mathbb{R}^N \to \mathbb{R}_0^+$	$\lVert \mathbf{x} \rVert_p = \left(\sum_i \lvert \mathbf{x}_i \rvert^p\right)^{\frac{1}{p}}, 0 < p$
$\lVert \cdot \rVert_0, \ell_0 : \mathbb{R}^N \to \mathbb{R}_0^+$	$\lVert \mathbf{x} \rVert_0 = \#\{i, \mathbf{x}_i \neq 0\}$
$\lVert \cdot \rVert_\infty, \ell_\infty : \mathbb{R}^N \to \mathbb{R}_0^+$	$\lVert \mathbf{x} \rVert_\infty = \max_i \lvert \mathbf{x}_i \rvert$
$\lVert \cdot \rVert_F$	Frobenius norm, $\lVert \mathbf{A} \rVert_F = \left(\sum_{i,j} a_{i,j}^2\right)^{\frac{1}{2}}$
$\mathcal{J}(\cdot), \mathcal{J}_p(\cdot)$	sparsity measure
$\mathcal{J}_{p,q}(\Theta)$	mixed sparsity measure
$f_Y(y)$	pdf of random vector Y at an instance y
$\mathcal{L}(\cdot)$	likelihood function and Lagrange function

Chapter 1
Introduction

1.1 Background

Signal processing received significant attention in the last century as it found numerous applications in military and non-military products. The products dealing with audio, image, video, sonar and radar are only some examples of these products. The processing was started in the analogue domain at the beginning. Although there still exist products based on the analog processing of signals, most of the new products use an analog to digital converter followed by digital processing of signals, which reduces the overall product costs and improves the efficiency of the products.

This thesis is about discrete signal processing using a new technique, called sparse coding. The aim in sparse coding is to (approximately) represent the original discrete signal minimally, i.e most of the coefficients in the new representation are zero and the signal is presented using only a few coefficients. Using a minimal representation can improve the performance of some signal processing algorithms for different applications. Although some parts of this thesis are introduced for *sparse source coding*, the main focus is about general sparse coding, which can be used for different applications.

This chapter briefly introduces the sparse coding and the model selection problems while exploring the challenges of this model.

1.1.1 Sparse Coding

The minimal representation of the natural signals is not a new idea. It has been used in "Transform Coding" for decades, see [Mal99] for a more detailed review on transform coding. In this framework, we use an orthogonal transform to represent a signal with few non-zero coefficients. In general, it is impossible to find a linear transform for which a given class of signals has such an exact minimal representation. Fortunately it is often possible to approximately

represent almost all signals from a class with a small set of coefficients. The Fourier-type transforms have successfully been used as the orthogonal transform. The idea of using a minimum time-frequency spread transform was presented by Gabor [Gab46]. The elementary functions, called atoms, in this framework are not orthogonal and we often have an overcomplete set of elementary functions, which is called a dictionary. This opened a new window to the overcomplete signal representations and approximations. As a result, some researchers, see [Dau92] and references therein, introduced a new class of signal representation methods, called the frame method. When the dictionary is overcomplete, the representation of the input signal is not unique. The frame method represents the signal with a minimum ℓ_2 norm. This representation is useful for some applications in which the resilience or robustness of the representation is required [Cve03, GKK01, GVT98]. An important disadvantage of the frame method, which is the most important reason for it to not being used for coding applications, is its non-minimal representation. In contrast, sparse coding was proposed to find an overcomplete representation, similar to frame method, which is also in some sense minimal. Unfortunately, it can not be done using a linear operator and it has been shown that sparse coding in an overcomplete setting is an NP-hard problem in general [DMA97, Nat95]. Informally, this means that there do not exist any tractable algorithms to solve it in general.

Different greedy methods have been proposed to find a sub-optimal solution [MZ93, PRK93, DMA97, Nat95, CBL89, Tem03, DTDS06]. These algorithms gradually increase the approximation precision by refining the set of selected atoms starting with an inaccurate approximation and, in each iteration, adding one or more coefficients to the set of non-zero coefficients. Some have an extra operation which updates some of the previously selected coefficients [PRK93, DMA97, BD08a]. These methods are among the fastest sparse coding methods. Some structures in the dictionary help us to implement the algorithm faster. Among these structures, dictionaries which are shift invariant or union of basis structures have received special attention [BD06, GN03]. The performance of greedy sparse approximation methods has been studied [Tem03, Tro04a] and conditions under which they recover a sparse signal have been established. It has also been shown that there are upper bounds for the approximation errors of them which decay exponentially [GV06, Tro04b] .

The second class of sparse approximation methods is based on minimization of a sparse penalty subject to a linear or quadratic constraint. These constraints are related to approximation error. The ℓ_p, where $p \leq 1$ has often been used as sparsity penalty [KRE$^+$99]. However for $p < 1$

these optimization problems are not convex. Interestingly, the ℓ_1 sparse coding is a convex problem, which can be solved using standard optimization methods, see for example [BV04]. Because of these unique features of the ℓ_1 norm, ℓ_1 sparse coding has been deeply studied theoretically and practically [Tro06b, DE03, GN03]. Numerous practical algorithms have been introduced to effectively find the solution [CDS98, GR97, RK99, DDD04, DDFC08, FNW07, KKL$^+$07, EMZ07, CW05]. They can be classified as linear and quadratic programming, e.g. [CDS98, KKL$^+$07], and (sub) gradient descent based methods, e.g. [FNW07, EMZ07]. Although these methods are often slower than the greedy sparse coding methods, they are recommended for some applications, such as Compressed Sensing (CS) [CRT06a, Don06] and signal denoising [Don95, DJ94]. The ℓ_1 sparse coding is solveable with a polynomial time algorithm. It is in contrast with the ℓ_p sparse coding problem where $p < 1$ where there is no known non-combinatorial method available to find the global minimum. Analysis of the algorithms based on $\ell_p : p < 1$ is very difficult. In most of the few available reports the analyzes are based on the global minimum of the ℓ_p sparse coding problem [GN03, FL09]. In practice it has been observed that in many cases using $p < 1$ and reweighting technique makes the algorithms faster and also gives sparser solutions [GR97, RK99, DDFC08, FN05, CWB08]. For an actual algorithm analysis in this setting see [DG09].

Another class of sparse coding methods includes the algorithms which minimize the original sparsity measure, i.e. the number of non-zero coefficients. These methods are mostly based on the gradient (or alternated) projections of the exact/approximate representation constraints (or sets) and the set of fixed ℓ_0 or a constraint which is directly related to ℓ_0 [KR03, HGT06, BD08b, MP06]. Recently, modified versions of the gradient projection method have been introduced, which is based on gradually de-smoothed ℓ_0 sparsity measure [PM07, MBZJ09, MBJ08]. In the latter methods, the ℓ_0 has been relaxed in the beginning and gradually tends to ℓ_0, to prevent stopping in a *bad* local minima. Although there is no mathematical proof to show the advantages of these methods, simulation results show faster convergences and sparser solutions in some examples.

1.1.2 Dictionary Selection

In a sparse coding method first we should choose an appropriate dictionary. An easy way to generate an overcomplete dictionary is concatenating orthogonal bases. Although this is a simple method to generate the dictionary, promising results have been reported in different ap-

plications [DD06, RRD08b, ESQD05]. There are also some mathematical analyses on the exact sparse recovery [GN03, EB02] and a simple sparse approximation method [SBT00] using union of bases. Another class of pre-designed dictionaries are frames. For example, the undecimated wavelets [SED04] and the overcomplete (multiscale-) Gabor [MZ93, GB03] dictionaries are from this class. The structures of these pre-designed dictionaries provide fast implementations of the sparse coding methods. For example, when the signal size is n, the complexity of the implementation of Fourier-type, curvelet and bandelet dictionaries are $\mathscr{O}(n \log n)$ and wavelet, contourlet and steerable wavelet dictionaries are $\mathscr{O}(n)$. These are clearly more efficient than unstructured dictionaries, of complexity $\mathscr{O}(n^2)$.

A difficulty in using pre-designed dictionaries is that they are not adapted to the structures of a given class of signals. It is also important to use a dictionary which is optimized for an application. For example, when the sparse approximation is used for coding, different dictionaries are suitable for different bit-rates [RRD08a]. As an empirical solution the dictionary can be found using dictionary learning methods which sparsifies the approximations [OF97, EAH99a, LS00, KMR$^+$03, AEB06]. These methods use a set of training samples, which are put into the columns of a matrix, called the signal or input matrix. These methods often use an optimization technique called block-relaxation, see for example [Lee94]. In this framework we optimize based on each block of variables, here the coefficient matrix and the dictionary, while the other blocks remain fixed. The difference between these methods is in the parameter block selection and the optimization method. In the dictionary learning problem, it is also necessary to impose a constraint over the set of dictionaries to make the problem well-defined. Two often used constraints are fixed column and Frobenius norm dictionaries [KMR$^+$03]. In each step of block-relaxed minimization, we reduce a proposed objective function by updating the selected block of parameters. Because the dictionary learning problem is a non-convex optimization problem, a local minimum is yielded from a gradient descent method. For a given set of training samples, [AEB06] has shown under which conditions, the dictionary is unique, up to a permutation in the location and the sign of the atoms. These conditions are based on the signal sparsity and the number of the training samples.

Most of the dictionary learning algorithms reduce the total approximation error by updating the dictionary while keeping the sparsity fixed. The approximation error is zero in a sparse representation and the conventional dictionary learning algorithms are not applicable. A new dictionary learning method has recently been introduced in [Plu07a], which can be used for

dictionary learning in this framework.

Another type of signal adapted dictionary design method is introduced in which a parametric function is used to generate the atoms [SM08]. Here the parametric function is chosen such that it has similarities with the generative model or the human perceptual system [Dau80, PAG95]. The dictionary is perception adapted in this case. Other parametric dictionaries have also been used to induce some structures on dictionaries. The designed parametric dictionary can be used as an alternative frame, in the frame method, which often has tighter frame bounds. Although a strong similarity between these perceptual model and the learned dictionary has been observed [Lew02, SL05, SL06], there have not been investigated respectably.

It deserves mentioning that the dictionary learning can also be seen as a generalization of the conventional Blind Source Separation (BSS) [JH91], which deals with complete representations. Various methods have been introduced to solve BSS problem using different prior assumptions for the sources, e.g. Principal Component Analysis (PCA), see for example [Jol02] and references therein, and Independent Component Analysis (ICA) [Com94, HO00]. In this formulation, the dictionary is called the "mixing matrix".

The BSS problem can be generalized using an underdetermined model and assuming a pattern for sources. When the source is sparse, this problem is called Sparse Component Analysis (SCA) [ZP01, GTC05] which is strongly related to the dictionary learning problem. The SCA problem can be interpreted as an application of the dictionary learning, followed by sparse approximations of the sources.

1.1.3 Quantization

Computerized signal prosessing systems, particularly digital communication systems, operate on quantized valued signals, therefore the sparse approximation of the signals should be quantized to fit in this framework. There are two distinct approaches to find the quantized value sparse approximations (QVSA).

- *A posteriori quantization:* An easy way to find the QVSA is to quantize the coefficients *a posteriori* [NZ00, FV01, FVFK04, DD06]. Here a quantizer, which is designed based on the experimental probability density function (pdf) of the coefficients, is used. Non-orthogonality of the atoms makes the analytical optimization of such quantizers very dif-

ficult. [FVFK04] used an approximate bound on the rate-distortion (R-D) of the sparse coding and designed an optimal scalar quantizer, using the experimental pdf, for the coefficients. Another approach assumes prior pdf for the dictionary elements, for example Gaussian i.i.d., to find an analytical formula for the R-D [FRGR06].

Quantization of the coefficients can also cause other issues. As an example, the quantization of an overcomplete representation might be inconsistent[1]. The inconsistency is caused by non-orthogonality of the bases functions in an overcomplete representation, e.g. frame expansion and sparse representation. [GVT98] correspondingly introduced algorithms based on linear programming and alternating-projection for consistent quantizations of the representation using the frame method and Matching Pursuit algorithm.

- *In-loop quantization:* The sparse coding algorithm using in-loop quantization is a modified version of the iterative sparse coding algorithm, with an extra coefficient quantization step in each iteration [GV97, DZ03]. Let us assume the quantized sparse coding to be the sparse coding in the integer domain[2]. To find the quantized sparse approximation, each iteration can be relaxed by real valued optimization followed by a projection onto the admissible set, here the set of integer coefficients. For example if the sparse coding is based on the gradient descent method, the in-loop quantization is then based on the gradient-projection method.

It should be noted that the standard scalar quantization of an orthogonal representation is unique and consistent. The quantization error caused by quantization of each coefficient is then orthogonal to the quantization errors caused by the other coefficients. In this case in-loop quantization does not therefore decrease the total quantization error.

1.2 Contributions

Parts of this thesis have already been published in peer reviewed journals and conference proceedings. A list of these publications is as follows,

Peer Reviewed Journal Articles:

[1] Let $\mathbb{Q}$ and $\mathbb{A}$ be the quantization and the reconstruction operators respectively. A quantizer $\mathbb{Q}$ is inconsistent when $\mathbb{Q} \neq \mathbb{Q}\mathbb{A}\mathbb{Q}$

[2] This statement is valid only when the quantizer is a uniform scalar.

1. “Parametric Dictionary Design for Sparse Coding”, with L. Daudet, M. Davies, IEEE Transaction on Signal Processing, Vol. 57, No. 12, pp 4800-4810, 2009.
2. “Dictionary Learning for Sparse Approximations with the Majorization Method”, with T. Blumensath, M. Davies, IEEE Transaction on Signal Processing, Vol. 57, No. 6, pp 2178-2191, 2009.

Conference Proceedings:

1. “Structured and Incoherent Parametric Dictionary Design”, with L. Daudet and M. Davies, accepted for presentation in the IEEE International Conference on Acoustics, Speech and Signal Processing, 2010 (Invited Paper).
2. “Compressible Dictionary Learning for Fast Sparse Approximation”, with M. Davies, IEEE Workshop on Statistical Signal Processing, 662-665, Aug. 31- Sept. 3, 2009.
3. “Parsimonious Dictionary Learning”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing,pp. 2869-2872, April 2009.
4. “Parametric Dictionary Design for Sparse Coding”, with L. Daudet, M. Davies, Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09), 2009.
5. “Regularized Dictionary Learning for Sparse Approximation”, with T. Blumensath, M. Davies, European Signal Processing Conference (EUSIPCO), August 2008.
6. “Iterative Hard Thresholding and L_0 Regularisation”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing, 877-880, April 2007.
7. “Quantized Sparse Approximation with Iterative Thresholding for Audio Coding”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing, 257-260, April 2007.

Communications (without proceedings):

1. “Structured and Incoherent Parametric Dictionary Design”, with L. Daudet and M. Davies, INSPIRE Conference on information representation and estimation, London UK, September 2009.

2. "Compressible Dictionary Learning for Fast Sparse Approximation", with M. Davies, Workshop on Sparsity and its Application to Large Inverse Problems, Cambridge, UK, December 2008.

1.3 Organization

This thesis has two parts. In the first part, the sparse coding problem is formulated in Chapter 2 and various sparse approximation methods are surveyed in Chapter 3. This part is presented to introduce the readers to the sparse coding field and get them ready for the main contribution of thesis in the second part, i.e. dictionary selection methods. The methods are classified here based on the original optimization techniques, which emphasizes similarities and differences of them. The original contribution of the thesis in this part is limited to the quantized sparse approximation method using iterative hard thresholding, see section 3.5.

The second part starts with a survey on the state of the art dictionary learning methods in Chapter 4. The survey is organized based on the approach of the methods, and is thus not in a chronological order. The aim is to help the readers understand the relations between different algorithms. It furthermore introduces unstructured before structured dictionary learning methods as this facilitates understanding of the latter methods, where they often rely on the unstructured dictionary learning methods. Chapter 5 presents a new algorithm for unstructured dictionary learning, which has a convergence proof. It is based on a well-known optimization technique called the majorization minimization method. It is also very flexible and can use different sparsity measures in this framework. Two special cases have been investigated in the following chapters, 6 and 7. Chapter 6 applies a joint sparsity penalty, which will be defined in Chapter 2, to the dictionary learning to find a small size dictionary. Chapter 7 introduces a new generative model for the dictionary to facilitate the implementation of the dictionary in sparse coding. These modified dictionary learning problems solved using the same majorization minimization method, which is presented in Chapter 5.

Chapter 8 introduces a different approach to the dictionary selection. The proposed method, which is called parametric dictionary design, is an alternative to dictionary learning, where the domain knowledge is presented using a (set) of parametric generative function(s). The dictionary can be found by minimizing an objective that promotes the incoherence of the dictionary. A practical algorithm for dictionary design in this framework is also presented in this chapter.

This thesis is concluded in Chapter 9 where some directions for the future work are presented. Four appendices have been given here to complete the thesis. Appendix A extends the majorization function of Chapter 3 to the space of matrix value functions. Appendices B and D analyze the convergence of algorithms are presented in Chapter 5 and 8, respectively. The gradient of the Gammatone parametric dictionary, which is the case study of Chapter 8, is derived in Appendix C.

Part I

Sparse Coding

Chapter 2
Sparse Coding Formulations

2.1 Introduction

In this chapter, the sparse coding problem is formulated. One can classify sparse coding problems as sparse approximations and sparse representations. A sparse representation is sometimes called an exact sparse representation. We start by presenting different formulations for the problem. These formulations are then extended to the sparse coding in matrix form. This provides an extra flexibility to find matrices with different sparsity patterns along the column or the row directions. It will be shown in Part II that this formulation can be used for dictionary learning.

2.2 Sparse Coding Formulations

The aim of sparse coding is to represent a signal exactly or approximately by the minimum number of coefficients. Let $\mathbf{D} \in \mathbb{R}^{d\times N}$, $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the generator matrix (or dictionary [MZ93]), the signal and coefficient vectors respectively. The linear generative model is now formulated as,

$$\mathbf{y} = \mathbf{D}\mathbf{x}. \tag{2.1}$$

We assume that $\mathbf{D}$ is full rank ($\mathrm{rank}(D) = \min(d, N)$). In this framework when $d = N$, the exact coefficient vector is uniquely found by the inverse operator of $\mathbf{D}$, $\mathbf{x} = \mathbf{D}^{-1}\mathbf{y}$. When the model is over-determined $d > N$, one can choose a full rank $\mathbf{D}_r \in \mathbb{R}^{d\times d}$, by using d rows of $\mathbf{D}$, and find $\mathbf{x}$ by using $\mathbf{D}_r^{-1}$ matrix. The under-determined model ($d \leq N$), which is the main focus of this thesis, does not have a unique solution. This means that the number of equations are less than the number of unknown parameters. To resolve this ambiguity, different constraints have been proposed to impose prior information over the coefficients. The most well-known constraint is the minimum ℓ_2 norm, which has been used for decades. It can be interpretted as imposing a Gaussian assumption on the pdf of coefficients, which is an optimal

assumption for many applications. A minimum ℓ_2 norm representation can be calculated very fast with a *linear* operation. The inverse operator $\mathbf{D}^\dagger$ is called *pseudoinverse* and can be found by,

$$\mathbf{D}^\dagger = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}. \tag{2.2}$$

An issue with using minimum ℓ_2 representation is that the coefficients are mostly non-zero. Although it is useful for certain applications, for example when we have erasure or noise [GKK01] in the model, the minimum ℓ_2 overcomplete representation is not the optimal representation for a significant class of signal processing applications. Instead, one can use a sparsity penalty $\mathcal{J}(.)$ and find the sparsest representation, see for example [KRE+99] and [CDS98] and references therein. The signal representation is then formulated by,

$$\min_{\mathbf{x}\in\{\tau:\mathbf{y}=\mathbf{D}\tau\}} \mathcal{J}(\mathbf{x}). \tag{2.3}$$

In the ideal case, the operator $\mathcal{J}(.)$ counts the number of non-zero components. However the optimization problem (2.3) using such a sparsity measure, which is called ℓ_0, is an NP-hard problem, in general [DMA97]. Finding the solution for this type of problem is computationally difficult, even in a medium size problem, and it can only in general be done using an exhaustive search. Another approach is to apply an optimization technique to reduce ℓ_0, subject to the constraint proposed in (2.3) [KR03, BD08a], which can only find a sparser representation than the initial solution. Alternatively a series of smoothed objectives, which converge to ℓ_0 in the limit, can be optimized iteratively [MBZJ09]. In practice better local minima are observed using the smoothed objectives.

To find an acceptably sparse representation, one can use a relaxed sparsity measure. The relaxed sparsity measure is not necessarily smooth and is often fixed during sparse approximation. An often used relaxed $\mathcal{J}(.)$ is $\ell_p^p(\mathbf{x}) := \sum_{1\leq i\leq N} |x_i|^p$, where x_i is the i^{th} element of $\mathbf{x}$ and $p \leq 1$. A special case, where $p = 1$, is particularly interesting since the problem (2.3) for $p = 1$ is convex and can be solved using different convex optimization methods. The global minimum[1] is then found using these optimization methods. Furthermore, the analysis of the optimization *methods* are easier using ℓ_1 sparsity measure. The sufficient conditions, under which the solutions of the sparse representation using ℓ_1 and ℓ_0 are equivalent, are investigated in [Don04a, GN07].

[1] Because the objective is not strictly convex, it could have non-unique solutions. Under a mild condition, which is often satisfied by the sparse representation settings, the solution is unique.

The set of K-sparse vectors can have infinitely large members in a norm space. Let the set of K-sparse signals has an upper bound on the components, i.e. $\|\mathbf{x}\|_\infty < c$. The use of ℓ_1 is justified by showing that the ℓ_1 objective $\frac{1}{K}\|\mathbf{x}\|_1$ is the "convex envelope" [BV04] of the non-convex ℓ_0 [Dat09]. Therefore, there is no better convex approximation for an ℓ_0 objective in this sense. Using a more accurate approximation for the objective, leads to a non-convex optimization problem. Various methods for optimizing such an objective have been introduced [GR97, RK99, CWB08, DDFC08]. Although there is no easy way to exactly solve the sparse representation problem using this class of sparsity measures, in practice the sparse vectors found by these methods are sparser than ℓ_1 sparse representation. A slightly different sparsity measure to the class of ℓ_p sparsity measures, is the logarithmic sparsity measure. It has some useful properties which facilitate the minimization.

$$\mathcal{J}_{log}(\mathbf{x}) = \sum_{1 \leq i \leq N} \log x_i^2 \tag{2.4}$$

This is sometimes called *Gaussian entropy* [KRE+99, RK99].

The exact sparse coding problem introduced in (2.3) is for a noise-free model. In practice, it is often important to consider the effect of noise effect in the model. The noise is often introduced as an additive term. The signal generative model is then presented by,

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}, \tag{2.5}$$

where $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{D}$ are as before and $\mathbf{n}$ is the noise vector. Based on the distribution of noise in the model (2.5), one can define a measure on the signal space. When the noise has Gaussian or Laplace distribution, the expectation of the noise can empirically be calculated using ℓ_2 or ℓ_1 norms respectively. The ℓ_2 norm has often been used in the sparse coding problem and from now on, we will use the ℓ_2 norm as the measure of error. One can also assume the model mismatch as the noise in the proposed model in (2.5). In this framework, an underdetermined signal *approximation* can be formulated by,

$$\mathbf{x} \in \{\forall \theta : \|\mathbf{y} - \mathbf{D}\theta\|_2 \leq \epsilon\} \tag{2.6}$$

where ϵ is a constant. The problem is the same as (2.1) using $\epsilon = 0$. (2.6) is also an underdetermined system and the solution space has more than one element. By minimizing a strictly convex objective, e.g. $\ell_p : 1 < p$, over this convex set we can find the *unique* solution. The

minimum ℓ_2 overcomplete approximation has been used for denoising, parameter estimation, system identification and classification. The minimization of the ℓ_2-norm over (2.6) can be solved analytically using the *regularized pseudoinverse* operator defined by,

$$\mathbf{D}^{\dagger} = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \epsilon^2 I)^{-1}. \tag{2.7}$$

This operator is preferred over (2.2) in practice, not only because it considers the noise effect, but also because it can solve the ambiguity caused by any singularity of $\mathbf{D}\mathbf{D}^T$, when $\mathbf{D}$ is rank deficient.

Like the noise-free model, the *linear* operator (2.7) generally finds a non-sparse solution. A sparsity measure can be minimized, with the constraint (2.6) to find a sparser approximation. Although ℓ_1 is not strictly-convex, it can be shown that the following optimization problem, called Basis Pursuit DeNoising (BPDN), has a unique solution,

$$\min_{\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \leq \epsilon} \|\mathbf{x}\|_1. \tag{2.8}$$

The dual representation of BPDN, called LASSO [Tib96], is defined by,

$$\min_{\|\mathbf{x}\|_1 \leq \tau} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \tag{2.9}$$

Where there is an injective mapping between ϵ and τ such that BPDN and LASSO have the same solutions. These two problems are convex and can be solved exactly by using an appropriate convex optimization method. The solutions of BPDN and LASSO are sparse and denoised[2].

Sometimes it is useful to extend the problems (2.8) and (2.9) by using another sparsity measure. Although the problem is no longer convex, the local solutions, which can be found using some of the algorithms presented in Chapter 3, are often sparser.

(2.8) and (2.9) are constrained optimization problems. There are many effective optimization methods which can only be applied to the non-constrained problems. By using the Lagrangian multipliers method, we can generate an unconstrained problem. The optimization problem is

[2]The optimality of the solutions using an orthogonal dictionary is guaranteed [DJ94]. This framework has been used in the overcomplete setting. Promising results have been reported, for example, in [FRGR06].

now formulated by,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{2.10}$$

where λ is the Lagrangian multiplier. The sparsity of the approximation can be modified by changing λ. Although this optimization problem is not strictly-convex, it has a *unique* solution [CW05, Tro06b, Proposition 3.1]. This can be proved by showing that the quadratic part is strictly-convex, the remaining part $\|\mathbf{x}\|_1$ is convex and the objective is unbounded when $\|\mathbf{x}\| \rightarrow \infty$ [Zal02, Proposition 2.5.6]. The uniqueness of the solution is a necessary requirement for the Perfect (Exact) Recovery Problem [DH01]. It has been shown that the sparse representation of a signal is unique if the signal is sparse enough and the dictionary satisfies the Exact Recovery Condition (ERC)[3].

This change in definition significantly increases the number of algorithms that can be applied to solve the problem. For example most of the (sub-)gradient descent methods can now be applied to (2.10), see [FNW07, EMZ07]. Therefore (2.10) is the most desirable formulation for the sparse approximation problem.

Similar to the sparse representation problem, one can generalize the sparse approximation problem by using a different $\mathcal{J}(.)$ as the sparsity measure. The generalized form of (2.8) is formulated by,

$$\min_{\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \leq \epsilon} \mathcal{J}(\mathbf{x}), \tag{2.11}$$

and the generalized form of (2.10) is formulated by,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\mathcal{J}(\mathbf{x}). \tag{2.12}$$

If $\mathcal{J}(.)$ is non-convex, e.g. $\ell_p : p < 1$, sparse approximation problems (2.11) and (2.12) have numerous local minima and the global solution can not easily be found, in general[4]. In practice, it is observed that (2.12) for $\ell_p : p < 1$ often converges faster and/or finds sparser solutions [RK99, CWB08, DDFC08].

The solutions of the problems (2.11) and (2.12), when $\mathcal{J}(.) \neq \ell_0$, are always biased [CDS98]. It means there are better approximations with the same sparsity pattern. This can be compen-

[3]ERC of a set of indices Λ is defined by $ERC(\Lambda) := 1 - \max_{\omega \notin \Lambda} \|\mathbf{D}_\Lambda^\dagger \mathbf{d}_\omega\|_1$, where $\mathbf{D}_\Lambda$ is the matrix generated using the atoms indexed by Λ [Tro06b].

[4]We use the term *in general* to note that under certain conditions (2.12) and (2.10) share the solution support. In this case, the solution support of (2.12) could obviously be found by solving (2.10). An extra step is needed to find the coefficient magnitudes by solving a reduced order optimization problem, which has a unique solution.

sated using a post processing step called de-biasing. In this process the signal is orthogonally projected onto the space selected by the non-zero coefficients. Let $\mathbf{D}_I$ be the dictionary composed by using the selected atoms in the approximation. The orthogonal projection can be found using the linear operator pseudoinverse, which is already defined in (2.2). Because $\mathbf{D}_I$ depends on the sparsity pattern, the calculation of $\mathbf{D}_I^\dagger$ can not be done a priori. This error is often reduced using a sparsity measure which is closer to ℓ_0. This is also another reason that ℓ_p and logarithmic sparsity measures are preferred to be used in some practical applications of the sparse approximations [DD06, YBD07].

2.3 Sparse Matrix Coding

This section generalizes the sparse coding problem from the vector space to the matrix space. Let $\mathbf{Y} \in \mathbb{R}^{d\times L}$, $\mathbf{X} \in \mathbb{R}^{N\times L}$ and $\mathbf{D} \in \mathbb{R}^{d\times N}$ be the signal matrix, the coefficient matrix and the dictionary, respectively. When $d < N$ and $\mathbf{D}$ is full-rank, the underdetermined linear generative model is defined by,

$$\mathbf{Y} = \mathbf{D}\mathbf{X}, \tag{2.13}$$

and the noisy linear generative model is also defined by,

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{N}, \tag{2.14}$$

where $\mathbf{N} \in \mathbb{R}^{d\times L}$ is the noise (or model mismatch) matrix. Given $\mathbf{Y}$ and $\mathbf{D}$, the solution spaces for the problems (2.13) and (2.14) are respectively defined as:

$$\Lambda_{exact} := \{\forall \mathbf{\Theta} : \mathbf{Y} = \mathbf{D}\mathbf{\Theta}\}, \tag{2.15}$$

and

$$\Lambda_{noisy} := \{\forall \mathbf{\Theta} : \|\mathbf{Y} - \mathbf{D}\mathbf{\Theta}\|_F \leq \epsilon\}, \tag{2.16}$$

where $\|.\|_F$ is the Frobenius norm[5] and $\epsilon \in \mathbb{R}^+$. These convex sets have more than one element each, as a result of underdetermination of the generating system. We can now impose extra

[5]Frobenius norm is ℓ_2 norm of the matrix vector space and defined by $\|\mathbf{X}\|_F = |\langle \mathbf{X}, \mathbf{X}\rangle|^{1/2}$, where $\langle ., .\rangle$ is the inner-product of the matrix space which is defined by $\langle \mathbf{X}, \mathbf{Y}\rangle := \mathrm{tr}\{\mathbf{X}^T\mathbf{Y}\}$

constraints on the model to find desired solutions. Let the ℓ_p norm, for $p \geq 1$, be defined by,

$$\ell_p(\mathbf{\Theta}) = (\sum_{i,j} |\theta_{i,j}|^p)^{1/p}, \tag{2.17}$$

where $\theta_{i,j}$ is the (i,j) element of $\mathbf{\Theta}$. ℓ_p is a norm in the matrix space therefore $B_{\ell_p}(\gamma) = \{\mathbf{\Theta} : \ell_p(\mathbf{\Theta}) \leq \gamma\}$, called the ℓ_p ball, is closed and convex. Using a minimum ℓ_p constraint over Λ_{exact} and Λ_{noisy}, the cardinality of the solution sets are reduced to one, which is a similar result to the vector form of sparse approximation. A special case of this problem is when $p = 2$, where the solution can be found using the *linear* operator introduced in (2.2). (2.17) for a $p < 1$ generates a non-convex objective and (2.17) is no longer a norm. Similar to the vector space, $\ell_p^p(.) : p \leq 1$ generates a sparsity measure for the matrix vector space by the following formula,

$$\mathcal{J}_p(\Theta) = \sum_{i,j} |\theta_{i,j}|^p, \tag{2.18}$$

The sparse matrix representation or approximation are then defined by minimizing $\mathcal{J}(\mathbf{X}) = \mathcal{J}_p(\mathbf{X})$, such that $\mathbf{X}$ be in Λ_{exact} or Λ_{noisy} respectively. The sparsity measure (2.18) is an elementwise operator. In Chapter 3 it will be shown how this separability facilitates sparse matrix coding.

A variation of sparse matrix approximation can also be formulated using a Lagrangian multiplier λ as follows,

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \mathcal{J}(\mathbf{X}). \tag{2.19}$$

An advantage of the formulation (2.19) over minimizing $\mathcal{J}(.)$ over Λ_{noisy}, is that when $\mathcal{J}(.)$ is a column-wise operator, e.g. (2.18), it can be minimized column by column, using a standard sparse approximation method.

No sparsity pattern is proposed in the definition of $\mathcal{J}(.)$ in (2.18). That is the value of $\mathcal{J}(.)$ does not change by relocating the non-zero elements. Such a pattern is often desireable when natural signals are sought. Simultaneous sparse, tree and harmonic structures are some examples of such a sparsity pattern. In this framework a matrix $\mathbf{X}$ with a minimum number of non-zero columns is sought. The following definition for $\mathcal{J}(.)$ has been used for the simultaneous sparse coding [CREKD05, CH06, Tro06a, FR08b],

$$\mathcal{J}_{p,q}(\mathbf{\Theta}) := \sum_j (\sum_i |\theta_{i,j}|^q)^{\frac{p}{q}}, \tag{2.20}$$

where $0 < p \leq 1 \leq q$. By letting $q \geq 1$, $\mathcal{J}_{p,q}(\mathbf{\Theta}) = \sum_j (\|\theta_j\|_q)^p$. A minimum non-zero columns can be found by choosing $0 < p \leq 1$, which promotes the sparsity of $[\|\theta_j\|_q]_j$. Although it is possible to use any $q \geq 1$, particularly $q \to \infty$, it is preferred in practice to use $q \in \{1, 2\}$, which also provides noise robustness. Note that the sparsity measure $\mathcal{J}_{p,q}(.)$ defined in (2.20) is not an element-wise operator, when $p \neq q$. Here the conventional sparse approximation methods can not directly be used for this problem. The dictionary learning problem, using such a sparsity measure, will be presented latter and an efficient algorithm will be introduced to solve (2.19).

2.4 Summary

The sparse coding problem was formulated in this chapter by introducing some sparsity measures and the related optimization problems which should be minimized. In this framework, we constrain the solution space of an underdetermined linear system to the solutions with the minimal non-zero coefficients. The formulations were extended to the matrix vector space. This was done to facilitate the sparse coding of a set of signals or to induce a structured sparsity pattern within the matrix. The sparse coding formulations of this chapter are used in Chapter 3, where the sparse coding algorithms are discused and in Part II, where the dictionary learning problem is formulated.

Chapter 3
Sparse Coding Algorithms

3.1 Introduction

The sparse coding problem was formulated in Chapter 2, in which the aim is to minimize an objective, subject to a constraint. The constraint can be removed when the coefficients have only to be admissible. Different optimization methods have been introduced to solve (2.3),(2.11) and (2.12). The size of sparse coding problems is often such that some optimization techniques are not tractable. Although some linear/quadratic programming and stochastic sampling methods are tractable for small and medium size problems, they are too slow for the large problems. In contrast, the gradient descent based methods, which might not be fast enough for small problems, are good options for large scale sparse coding problems.

Direct optimization of the sparse coding problem is not the only way to find sparse codes. In practice it is sometimes preferred to solve the optimization problem using a greedy method. These greedy methods gradually increase the selected support of the coefficient vector to reduce the approximation/representation errors. These methods are especially useful when the size of the problem is large such that applying other optimization methods are not practical.

This chapter briefly reviews the sparse coding algorithms. These algorithms are numerous and it is difficult to completely cover them in a single chapter of a thesis. It is thus preferred firstly to classify different algorithms based on their approaches to the problem, then a brief explanation about the motivations and the applications of the approach are presented. The approaches, which are more often used by the researchers, are explored in more detail.

An optimization technique, called the majorization minimization method, has been shown to be very useful to solve the sparse approximation problem. This algorithm simplifies the problem by decoupling the multivariable optimization problem to some single variable problems. The decoupled problems can now be solved based on each individual coefficient. Because this technique will also be used in the following chapters, for dictionary learning, it is introduced here in more detail.

If the sparse approximation is subjected to a constraint on the coefficients, the optimization problem should then be modified slightly to handle such constraints. One of the constraints is quantization, i.e. the coefficients lie in the quantized value domain. A modification has been proposed which is the iterative quantization of the coefficients in each step of the sparse approximation method. A novel method is presented here to find quantized sparse approximations, which is a modification for the iterative hard thresholding algorithm [BD08b] with an in-loop quantizer [YBD07].

The sparse approximation methods are explored in the following section by starting with an overview on different approaches. Some greedy and gradient descent based methods are then introduced with an introductory presentation of the majorization minimization method. This technique, which is a special case of gradient projection method, is the basis of most fast gradient descent methods. Recently another technique, called proximal method [CW05], has also been presented to accelerate ℓ_1 sparse approximation problem. This is briefly introduced thereafter. Finally the quantized sparse approximation is introduced in Section 3.5.

3.2 Sparse Approximation Methods

Sparse coding methods can be classified based on their approaches to the problem. Some of these classes are as follows,

1. *Greedy pursuit:* These methods start with a coarse approximation and gradually refine the approximation by changing the selected set of atoms and the magnitudes of the selected coefficients. These methods include Matching Pursuit (MP) [MZ93], Orthogonal MP (OMP) [PRK93, DMA97] and their variations like Optimized OMP (OOMP) [RL02], Gradient Pursuit (GP) [BD08a], Stagewise OMP (StOMP) [DTDS06]. Slightly different methods in this class are the greedy methods for convex relaxed sparse approximation (2.10), called polytopes faces pursuit [Don04b, Plu06, Plu07b].

2. *Convex and non-convex optimization:* All methods that *directly* minimize the problems (2.8) and non-convex version of that, (2.11) or (2.12). When the problem is convex, the sparse codes can be found using linear and quadratic programming [BV04, CDS98][1]. These methods are not very efficient for large-scale problems. Other optimization meth-

[1] Some Matlab® implementations of such methods can be found in the following packages: 1- *Atomizer*, `http://sparselab.stanford.edu/atomizer`, 2- ℓ_1 *Magic*: `http://www.l1-magic.org`

ods, like gradient descent based methods and regression methods, are often preferred for large scale problems, see for example [DDD04, EHJT04, FNW07, BT08]. The objective that we want to minimize becomes non-convex using any non-convex sparsity measures, see Chapter 2. Finding the global minimum of such an objective is difficult in general. Some methods are proposed to find a local minimum of such optimization problems [GR97, RK99, CWB08, DDFC08]. It has practically been shown that the local minimum is often more sparse for the same approximation errors, which can justify the use of such methods [CWB08]..

3. *Based on Stochastic Modelling:* These methods are based on inducing some prior distributions onto the coefficient vectors, which promote sparsity of the representations. These methods are often based on the maximum a posteriori (MAP) framework, in which the Bayesian inference has been used to calculate the posteriori distribution, see for example [LS00, OF97, WR04]. These methods can generally be classified in the class of non-convex optimization methods.

4. *Exhaustive Search:* This method is only tractable when the size of problem is small or some prior information, for example about the support of coefficient vectors or the subspace in which the signal lies, is given. The complexity of the problem can be reduced using cutting-plane technique [TW09].

Although this classification is neither rigid, while some methods might fit into more than one classes, nor complete, while some methods do not lie on any classes, it gives us a perspective of the sparse coding methods. Some the frequently used convex/non-convex optimization methods and greedy methods have been explored in this chapter. The exhaustive search methods will not be reviewed in this thesis and some of the stochastic modelling based methods will briefly be explored in Chapter 4, when such stochastic models are used in the dictionary learning.

3.3 Greedy Methods

Greedy methods are introduced to find an acceptable sparse approximation using an iterative scheme. In each iteration of the algorithm, some atoms are entered in to the support by choosing non-zero coefficients and the values of coefficients are updated, which is the forward step, and then some atoms might be deselected from the support, which is the backward step. In the simpleset case one atom is added to the current support in the forward step and the backward

step keeps the coefficient values unchanged. This process can be extended by applying different forward and backward steps. The most famous method, called matching pursuit (MP) [MZ93], is inspired by the greedy regression methods. Because MP is simple to implement and it is very fast, it has been investigated in detail, see for example [Tro04a]. Some variations of MP are reviewed in [BD08a], and their computational complexity are compared. Some of these methods are introduced in the following.

3.3.1 Matching Pursuit

MP was initially introduced to find time-frequency representations of the signals in [MZ93][2] and was then found to be a very efficient sparse approximation method. The forward step of MP is to add one atom to the currently selected atoms. In a normalized dictionary $\mathbf{D}$, let $\{\alpha_i\}_{i\in[1,n]}$ be the selected atom indices and the signal $\mathbf{r}^{[n]} = \mathbf{y} - \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i} x_{\alpha_i}$ be the residual of $\mathbf{y}$ in the nth iteration. The atom which has the maximum correlation, i.e. maximum inner-product with the residual signal at the nth iteration, is selected as the $n+1$th atom. The atom selection step can be formulated as,

$$\alpha_{n+1} = \arg\max_i \left| \left\langle \mathbf{d}_i, \mathbf{r}^{[n]} \right\rangle \right|, \tag{3.1}$$

and the corresponding coefficient is found by the following formula,

$$\mathbf{x}_{\alpha_{n+1}} = \left| \left\langle \mathbf{d}_{\alpha_{n+1}}, \mathbf{r}^{[n]} \right\rangle \right|. \tag{3.2}$$

There is no backward step in MP to cancel out the atoms. MP terminates after a certain number of iterations or when the residual error $\|\mathbf{r}^n\|_2^2$ becomes small ($< \epsilon : \epsilon \in \mathbb{R}^+$). An issue with MP is that the algorithm might select an already selected atom, which makes the convergence of the algorithm slow. If the aim is to find a quantized approximation of the signal, the selected coefficient can be quantized at each iteration [DZ03]. The quantization error might be compensated by the following selected atoms, as long as the following selected atoms are non-orthogonal to the current atom.

Another issue with MP is that the coefficients do not provide the best approximation using the selected support. This can be compensated by orthogonally projecting the signal onto the span

[2] This greedy method was originally introduced in the high resolution radio interferometry, called the algorithm CLEAN [H74], to induce sparsity on representations. MP is more often used in the applied and computational harmonic analysis to refer the same algorithm.

of the support. It is the motivation for another greedy algorithm, which will be explored in the following subsection, called Orthogonal MP.

3.3.2 Orthogonal Matching Pursuit

Using the coefficient selection step (3.2), we can easily show that $\mathbf{d}_{\alpha_{n+1}} \perp \mathbf{r}^{[n+1]}$. This fact might not be true for all $\{\mathbf{d}_{\{\alpha_i\}_{i\in[1,n]}}\}$ and $\mathbf{r}^{[n+1]}$. Let $\mathbf{r}^{[n+1]} = \mathbf{r}_O^{[n+1]} + \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i$ such that $\forall i \in [1, n+1] : \mathbf{d}_{\alpha_i} \perp \mathbf{r}_O^{[n+1]}$. In other words, $\{\beta_i\}_{i\in[1,n+1]}$ is found by projecting $\mathbf{r}^{[n+1]}$ onto $\text{span}\{\mathbf{d}_{\alpha_i}\}_{i\in[1,n+1]}$ and $\mathbf{r}_O^{[n+1]}$ is found by $\mathbf{r}^{[n+1]}$, with subtracting the projection. A relation between $\|\mathbf{r}^{[n+1]}\|_2^2$ and $\|\mathbf{r}_O^{[n+1]}\|_2^2$ can be found, using the orthogonality of $\mathbf{r}_O^{[n+1]}$ and $\sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i$, as follows,

$$\begin{aligned} \|\mathbf{r}^{[n+1]}\|_2^2 &= \|\mathbf{r}_O^{[n+1]} + \sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i\|_2^2 \\ &= \|\mathbf{r}_O^{[n+1]}\|_2^2 + \|\sum_{i\in[1,n]} \mathbf{d}_{\alpha_i}\beta_i\|_2^2, \\ \therefore \|\mathbf{r}_O^{[n+1]}\|_2^2 &\leq \|\mathbf{r}^{[n+1]}\|_2^2. \end{aligned} \tag{3.3}$$

This motivates us to apply back projection to reduced the residual. Orthogonal MP has been introduced in [PRK93] and [DMA97] in such a framework. Therefore an extra operation, in forward step of OMP, is orthogonal projection of the signal onto the space which is specified by currently selected atoms. Although this step is computationally expensive, it can be implemented more efficiently using QR and Cholesky matrix factorizations, see for example [Tro04a] and [BD08a] for more detail. However the back projection operator is not really tractable for large scale problems. The gradient pursuit algorithm was introduced in [BD08a] to relax the backward step and reduce the computational complexity of the algorithm. This algorithm is explored in the following subsection.

3.3.3 Gradient Pursuit

The extra step of OMP includes an orthogonal projection onto the span of the selected atom. This projection can be done using pseudoinverse operator which was defined in 2.2. A matrix inversion is needed to apply this operator, which is computationally expensive in a large scale problem. Although there are some more efficient ways to calculate pseudoinverse of such matrices using their structures [BD08a], an alternative can be to relax the coefficient adjustment

step. Instead of fully projecting the residual onto the selected space, we can choose a new coefficient vector, with the same support, with less residual error. Let the residual error at the $n+1^{th}$ iteration be noted by $\mathbf{r}_R^{n+1}$. A new *relaxed* OMP would be relevant if the residual satisfies the following inequality,

$$\|\mathbf{r}_O^{[n+1]}\|_2^2 \leq \|\mathbf{r}_R^{[n+1]}\|_2^2 \leq \|\mathbf{r}^{[n+1]}\|_2^2. \tag{3.4}$$

In other words, the coefficient adjustment step is to reduce the following cost function, by changing $\{x_{\alpha_i}\}$,

$$\|\mathbf{y} - \sum_{i \in \mathcal{I}} \mathbf{d}_{\alpha_i} x_{\alpha_i}\|_2^2, \tag{3.5}$$

where $\mathcal{I}$ includes the indices of all selected atoms, up to the $n+1^{th}$step , and $|\mathcal{I}| \leq n+1$. The minimizer of (3.5) is the projection onto $\text{span}\{\mathbf{d}_i\}_{i \in \mathcal{I}}$, which can be found using the gradient descent or the conjugate gradient methods. The Gradient Pursuit method uses a certain number of iterations of these iterative algorithms [BD08a], which are also guaranteed to satisfy (3.4).

The greedy algorithms are numerous and interested readers can refer to [TW09] for a more detailed review. In the next section, some of the methods for minimizing the relaxed sparse approximation problems are explored. These methods are mainly based on the iterative update of solutions in the negative gradient direction. Alternatively, the update of the coefficient vector can be in the direction which minimizes a surrogate objective in a majorization minimization framework. This framework is introduced at the beginning of next section, as it will be used here for sparse approximation and in Chapters 5, 6 and 7 to update the dictionary.

3.4 Relaxed Sparse Approximation Methods

The sparse approximation (2.12) is called "relaxed", when the sparsity measure $\mathcal{J}(.) \neq \|.\|_0$. The objective of relaxation is to make the objective function continuous and piecewise differentiable. The optimization of such a problem is easier, as long as various (sub-) gradient methods can be used. If the relaxed objective is convex, the global minimum is found using a gradient descent method. Although this is no longer true for the non-convex objective, sparser solutions can often be found by warm starting[3] and using a suitable step size for each update.

[3]Initializing the algorithm with a point satisfying some conditions. Starting with the convex relaxed solution or another sparse solution are some examples of such a warm start.

One class of sparse approximation methods either explicitly or implicitly is based on an optimization technique called majorization minimization method. This framework helps to simplify a complex multivariable optimization problem to an iterative optimization of a set of single variable optimization problems, which can be optimized independently. This framework is explained in the next subsection, which is followed by introducing the sparse approximation methods based on this technique.

3.4.1 Majorization Minimization Method

Optimization of a multivariable problem like (2.12) is challenging. A technique, called "Majorization Minimization Method" [Lee94, LHY00], will be introduced here to simplify such problems in an iterative framework. In the majorization method, the objective function is replaced by a surrogate objective function which majorizes it and can be easily minimized. Here we are particularly interested in surrogate functions in which the parameters are decoupled, so that the surrogate function can be minimized element-wise.

A function ψ majorizes ϕ when it satisfies the following conditions,

$$\begin{aligned} \phi(\omega) &\leq \psi(\omega, \xi),\ \forall \omega, \xi \in \Upsilon \\ \phi(\omega) &= \psi(\omega, \omega),\ \forall \omega \in \Upsilon, \end{aligned} \tag{3.6}$$

where Υ is the parameter space. The surrogate function has an additional parameter ξ. At each iteration we first choose this parameter as the current value of ω and find the optimal update for ω.

$$\omega_{new} = \arg\min_{\omega \in \Upsilon} \psi(\omega, \xi) \tag{3.7}$$

We then update ξ with ω_{new}. The algorithm continues until we find an accumulation point. In practice the algorithm is terminated when the distance between ω and ω_{new} is less than some threshold.

This iterative method can be viewed as a block-relaxed minimization of the joint objective $\psi(\omega, \xi)$ [Lee94]. In one step, we find the minimum of ψ based on ω. In the next step we

minimize the objective based on ξ.

$$\xi_{new} = \arg\min_{\xi \in \Upsilon} \psi(\omega, \xi) \tag{3.8}$$

In our formulation, minimization of $\psi(\omega, \xi)$ based on ξ is done using $\xi_{new} = \omega$ (due to the definition of majorization in (3.6)). We use this interpretation of the majorization method to show the convergence of the proposed method in Appendix B.

There are different ways to derive a surrogate function. Jensen's inequality and Taylor series have often been used for this purpose [Lan04, ZKY07]. The Taylor series of a differentiable function $\phi(\omega)$ is,

$$\phi(\omega) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{1}{2!} d^2\phi(\xi)(\omega - \xi)^2 + o(\omega^3). \tag{3.9}$$

When ϕ has a bounded curvature, i.e. $d^2\phi < c_s$ for a finite constant c_s, it is majorized by,

$$\phi(\omega) \leq \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2, \forall \omega, \xi \in \Omega, \tag{3.10}$$

and we can define $\psi(\omega, \xi)$ (which satisfies (3.6)) as follows,

$$\psi(\omega, \xi) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2. \tag{3.11}$$

Then, at each iteration, $\phi(\omega_{new}) \leq \psi(\omega_{new}, \omega) \leq \psi(\omega, \omega) = \phi(\omega)$, hence ϕ does not increase. Conditions for which these algorithms converge have been presented in [Lee94] and [Lan04].

In the next subsections some of the sparse approximation methods based on the majorization minimization method will be explored. The surrogate function can only be generated by a majorizing function for the quadratic term, the sparsity measure or both parts of (2.12). It demonstrates a possible wide range of sparse approximation methods, based on how the majorizing function is generated.

3.4.2 Iterative Thresholding

A difficulty in multivariable optimization problem like (2.12) is the coupling effect. It means the problem can not separately be solved with respect to each parameter. The sparsity measure

is often element-wise operator[4]. By majorizing the quadratic term of (2.12) with an element-wise objective, based on the coefficients, the new objective can be minimized element-wise. This has been applied to the sparse approximation problem, and called iterative thresholding[5] [DDD04, BD08b, FR08a]. The quadratic term of (2.12), $\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2$, has a bounded curvature and a majorizing objective can be found using Taylor series. By using (3.11), the majorizing objective for the quadratic is found as follows,

$$\begin{aligned} ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 &\leq ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 + c||\mathbf{x}-\mathbf{x}^\ddagger||_2^2 - ||\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}^\ddagger||_2^2 \\ &= ||\mathbf{y}-\mathbf{D}\mathbf{x}||_2^2 + \pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger) \end{aligned} \tag{3.12}$$

where $\pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger)$ is a function defined as follows,

$$\pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger) := c||\mathbf{x}-\mathbf{x}^\ddagger||_2^2 - ||\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}^\ddagger||_2^2. \tag{3.13}$$

If $c < \|\mathbf{D}\|$, where $\|.\|$ is the spectral norm operator, $\pi_{\mathbf{x}}(.,.)$ is a *convex* function based on $\mathbf{x}$, with a minimum at $\mathbf{x}=\mathbf{x}^{[n]}$. Let $\phi(\mathbf{x})$ be the objective in (2.12). $\psi(\mathbf{x},\mathbf{x}^\ddagger)=\phi(\mathbf{x})+\pi_{\mathbf{x}}(\mathbf{x},\mathbf{x}^\ddagger)$ satisfies the conditions (3.6). As mentioned in subsection 3.4.1, as long as the minimization of ϕ based on $\mathbf{x}^\ddagger$ is easily found by $\mathbf{x}^{\ddagger^*}=\mathbf{x}$, the alternating minimization can be done by minimizing ψ based on $\mathbf{x}$ and updating $\mathbf{x}^\ddagger$ by the current $\mathbf{x}^*$.

Although solving the decoupled problems is significantly easier than solving the original problem, only some of the sparsity measures $\mathcal{J}(.)$ lets the problem being solved analytically. Among them we are interested in ℓ_1 and ℓ_0[6], which will be presented in the next subsections. Although for the sparsity measure $\ell_p : p<1$, the decoupled problems can not be solved analytically, it can be solved using a gradient descent method to compare the results with the reweighting methods, which will be discussed in subsection 3.4.3.

[4]The joint sparsity measure is a column-wise operator which will be explored in Chapter 6, where a minimum size dictionary is sought.

[5]It is also called sparse approximations using majorization minimization method or Expectation Minimization (EM) based sparse approximations [FN03].

[6]Although the sparse approximation using ℓ_0 is not classified as the relaxed problem, it can be solved using MM technique [BYD07].

3.4.2.1 ℓ_1 relaxed sparse approximation

The sparse approximation in this setting was independently introduced in [FN03] and [DDD04]. The sparsity measure ℓ_1 is sum of the absolute values of coefficients, $\|\mathbf{x}\|_1 = \sum_{i\in[1,N]} |x_i|$. Let the auxiliary parameter $\mathbf{x}^{\ddagger}$ be $\mathbf{x}^{[n]}$. $\psi(\mathbf{x}, \mathbf{x}^{[n]})$ can now be reformulated as,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto c\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T \left(\mathbf{D}^T \left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda\|\mathbf{x}\|_1, \tag{3.14}$$

where $\propto$ means the equality, up to a constant. ψ is a convex function and its optimum can be found by the fact that sub-gradient should include zero, $\mathbf{0} \in \partial\psi(\mathbf{x}, \mathbf{x}^{[n]})$, where the sub-gradient $\partial\psi(\mathbf{x}, \mathbf{x}^{[n]})$ can be found by,

$$\partial\psi(\mathbf{x}, \mathbf{x}^{[n]}) = 2c\mathbf{x} - 2\left(\mathbf{D}^T \left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda\partial\|\mathbf{x}\|_1. \tag{3.15}$$

The optimal $\mathbf{x}^*$, which is the updated coefficients $\mathbf{x}^{[n+1]}$, can be found by applying the soft-shrinkage operator $\mathcal{S}_\lambda$ [DJ94] to the vector,

$$\mathbf{a} := \frac{1}{c}\left(\mathbf{D}^T \left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right). \tag{3.16}$$

$\mathbf{a}$ is actually a scaled gradient of the quadratic term, which is sometimes called the Landweber [Lan51] update [DDD04]. Soft-shrinkage is a non-linear operator defined by,

$$\{\mathbf{x}^{[n+1]}\}_i = \mathcal{S}_\lambda(\mathbf{a}) = \begin{cases} a_i - \lambda/2\ sign(a_i) & \lambda/2 < |a_i| \\ 0 & otherwise. \end{cases} \tag{3.17}$$

The convergence of the iterative method for ℓ_1 relaxed sparse approximation is shown in [DDD04]. The non-linear operator $\mathcal{S}_\lambda$ is the projection onto an ℓ_1 ball. The radius of the ℓ_1 ball can be calculated after projection. To accelerate the convergence of the sparse approximation Daubechies *et al.* [DFL08] suggested to adaptively change the radius of the ball. They also proved the convergence of the Gradient-Projection method with this setting.

Another algorithm, which is also based on Gradient-Projection method, is the algorithm proposed by Figueiredo *et al.* in [FNW07], called the Gradient Projection for Sparse Representations (GPSR). To simplify the problem and make the algorithm differentiable, they used a technique previously used in [CDS98], called a parameter splitting. In this method, each parameter is split to two positive parameters. Each pair of new parameters associates to an atom

and its negative version. The dictionary size thus becomes double in the new framework. (2.12) now becomes a constrained optimization problem with a differentiable objective as follows,

$$\min_{\mathbf{x},\bar{\mathbf{x}}\in\mathbb{R}_0^+} \|\mathbf{y} - \mathbf{D}\left(\mathbf{x} - \bar{\mathbf{x}}\right)\|_2^2 + \lambda \mathbf{1}^T\left(\mathbf{x} - \bar{\mathbf{x}}\right). \tag{3.18}$$

Figueiredo *et al.* proposed two different step sizes for the gradient projection method and proved the convergence of the final algorithm.

3.4.2.2 ℓ_0 sparse approximation

The sparsity measure ℓ_0 counts the number of non-zero coefficients and can be reformulated as $\|\mathbf{x}\|_0 = \sum_{i\in[1,N]} f(x_i)$, where,

$$f(\alpha) := \begin{cases} 0 & \alpha = 0 \\ 1 & otherwise. \end{cases} \tag{3.19}$$

Let the auxiliary parameter $\mathbf{x}^\ddagger$ be $\mathbf{x}^{[n]}$ as before. The surrogate objective is reformulated as,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto c\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\left(\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right) + \lambda \sum_{i\in[1,N]} f(x_i). \tag{3.20}$$

(3.20) is not convex and the sub-gradient method can not be used to minimize $\psi(\mathbf{x}, \mathbf{x}^{[n]})$. Instead we can decouple (3.20) to N optimization problems. The objective of the ith problem can be represented by,

$$\{\psi(\mathbf{x}, \mathbf{x}^{[n]})\}_i \propto cx_i^2 - 2x_i\left\{\mathbf{D}^T\left(\mathbf{y} - \mathbf{D}\mathbf{x}^{[n]}\right) + c\mathbf{x}^{[n]}\right\}_i + \lambda f(x_i) \tag{3.21}$$

(3.21) can be solved by letting x_i^* being zero or non-zero, followed by checking the validity of the solution. Let $\mathbf{a}$ be defined as in (3.16). x_i^* can be found using a non-linear operator $\mathcal{H}_\lambda$, called hard-shrinkage [DJ94], as follows,

$$\{\mathbf{x}^{[n+1]}\}_i = \mathcal{H}_\lambda(\mathbf{a}) = \begin{cases} a_i & \sqrt{\lambda} < |a_i| \\ 0 & otherwise. \end{cases} \tag{3.22}$$

The convergence of the iterative hard thresholding (IHT) is proved in [BYD07]. The algorithm can be modified to find a k-sparse approximation by replacing $\mathcal{H}$ with an orthogonal projection onto the space of k-sparse signals [BD08b]. That keeps the k largest coefficients and set the others to zero. This algorithm has also shown a promising performance in compressed sensing [BD09].

3.4.2.3 Other algorithms based on thresholding

The thresholding methods in the simplest for includes only one iterations with starting from $\mathbf{x}^{[0]} = \mathbf{0}$. These methods were explored in the fields of transform coding [Mal99] and statistical estimation [DJ94]. The simplicity of these algorithms is the main advantage over the iterative versions.

The iterative methods explained in the previous subsections are composed of two steps, one linear transform to find $\mathbf{a}$, followed by appling a non-linear operator. A drawback of thresholding algorithms is their slow convergence. Each step can be modified, for example by scaling, to improve the convergence rate of the algorithm [Ela06].

These algorithms can also be derived using a mathematical technique called operator splitting, see for example [CW05]. In this framework, the algorithm is composed of a forward and a backward operators which should alternatingly be applied to an initial solution. This framework allows us to use a double size walk, $\frac{2}{c}$ where $c < \|\mathbf{D}\|$, in the gradient direction before applying the soft shrinkage operator.

It has been shown that iterative soft thresholding converges R-linearly [7] in [BL08]. Bredies *et al.* [BL08] also showed that the asymptotic convergence rate is of order $\mathcal{O}(n^{-1})$[8]. By using an *"optimal first-order gradient method"*, also called the Nesterov's method [Nes83], the convergence rate can be improved to the order $\mathcal{O}(n^{-2})$ [Nes07, BT08, BBC09].

[7] Let $x^* = \lim_{n\to\infty}\{x^{[n]}\}$. It is said to converge to x^* at least with order $p \geq 1$, see for example [SM03], if there exists a constant c and a sequence $\{\epsilon_n\}$ such that $|x^{[n]} - x^*| < c\epsilon_n$ for all n and $\lim_{n\to\infty} \frac{\epsilon_{n+1}}{\epsilon_n^p} = \theta$ for $\theta \in (0, 1)$. A sequence is called to converge at least R-linearly if $p = 1$ [Pot89]. A similar definition can be presented on convergence of a sequence of vectors in a normed space.

[8] See [BD97] for the definition of Big $\mathcal{O}$ and Small o.

3.4.3 Iterative Reweighting

It was shown in the previous subsection that the majorization minimization method can be used to replace the quadratic term with some decoupled terms to facilitate the minimization. This technique can also be used to replace the sparsity measure with an ℓ_1 or ℓ_2 norm. Because there exist efficient algorithms to solve such a regularized approximation problem, the ℓ_p sparse approximation can easily be solved, i.e. finding a local minimum when $p < 1$, by iteratively solving majorized problem. This technique has also been known as iterative reweighting technique in literature. Some of these methods will be explained in the following.

3.4.3.1 Iterative Reweighted ℓ_1

ℓ_p for $p < 1$ is concave in each orthant. It can be shown that any concave function is majorized by the tangent line [Lan04], which can be used to generate a majorization function for the sparsity measure. If $\alpha \in \mathbb{R}^+$ and $\alpha_0 \in \mathbb{R}^+$, where α_0 is a fixed number, the following inequality holds,

$$\alpha^p \leq \alpha_0^p + p\alpha_0^{p-1}(\alpha - \alpha_0). \tag{3.23}$$

Note that such a majorizing function should be restricted to the corresponding orthant. One way is to use absolute value operator to restrict the majorizing line to the orthant in which current coefficient vector $\mathbf{x}^{[n]}$ is located and symmetrically duplicating that line in other orthants as follows,

$$\sum_{i\in[1,N]} |x_i|^p \leq \sum_{i\in[1,N]} |x_i^{[n]}|^p + p \sum_{i\in[1,N]} |x_i^{[n]}|^{p-1}(|x_i| - |x_i^{[n]}|). \tag{3.24}$$

When $\mathbf{x}_i^{[n]} \to 0$, the majorization function gets infinitely large, i.e. the original function is upperbounded by infinity, which is an obvious fact. In this case we let $\mathbf{x}_i^{[n]}$ stay at zero for the following iterations and reduce the size of problem. An alternative is to use a modified sparse approximation $\ell_{p,\epsilon}$ with $0 < \epsilon \ll 1$ as follows,

$$\ell_{p,\epsilon}(\mathbf{x}) = \sum_{i\in[1,N]} (|x_i| + \epsilon)^p. \tag{3.25}$$

(3.25) is bounded on $x_i \in \mathbb{R}$, which solves the singularity at $x_i^{[n]} = 0$. The majorizing function can now be found as follows,

$$\ell_{p,\epsilon}(\mathbf{x}) \leq \ell_{p,\epsilon}(\mathbf{x}^{[n]}) + p \sum_{i \in [1,N]} (|x_i^{[n]}| + \epsilon)^{p-1}(|x_i| - |x_i^{[n]}|). \tag{3.26}$$

By using such a majorization function for the sparsity measure we can find the surrogate objective as follows,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \sum_{i \in [1,N]} |w_i x_i|, \tag{3.27}$$

which can be solved in a weighted pursuit framework [DGV06]. Minimization of (3.27) would also be easier if we also majorize the quadratic part, and using the iterative thresholding scheme [FN05].

Iterative reweighted ℓ_1 has also been used for sparse representation with the ϵ-relaxed logarithmic sparsity measure $\sum_{i \in [1,N]} \log(|x_i| + \epsilon)$ in [CWB08].

3.4.3.2 Iterative reweighting ℓ_2

The surrogate objective made using a weighted ℓ_1 penalty is a close approximation of the original objective, i.e. the approximation error is small. A problem in using such a majorizing function is that the simplified problem is still difficult to solve, which can be solved by another convex relaxed sparse approximation method. An alternative is to majorize with a weighted ℓ_2, see for example [GR97], which simplifies the problem to a quadratic optimization problem and lets us to solve it analytically. In this framework the algorithm is sometimes called Iterative Reweighting Least Square (IRLS), but it only refers to a sub-class of algorithms in this class.

If the quadratic majorizing function for $\ell_p : p < 1$ satisfies following conditions, the optimization problem becomes more tractable.

1. *Decoupled,* to make the optimization easier.
2. *Even,* to follow the original objective, which is even.
3. *Has the same tangent space at* $\mathbf{x}^{[n]}$*:* to majorize $\ell_p \,|_{\mathbf{x}^{[n]}}$

The quadratic function which satisfies these conditions can be presented as $\sum_{i \in [1,N]} w_i x_i^2$,

where w_i's are some weights which can be found by [RK99],

$$w_i = |\mathbf{x}_i^{[n]}|^{2-p}, \tag{3.28}$$

and by [DDFC08],

$$w_i = (|\mathbf{x}_i^{[n]}| + \epsilon)^{2-p}, \tag{3.29}$$

for the ϵ-relaxed ℓ_p. If $\mathbf{x}_i^{[n]} = 0$ in (3.28), we let it to being zero in the following iterations. The surrogate objective can be found as follows,

$$\psi(\mathbf{x}, \mathbf{x}^{[n]}) \propto \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \sum_{i \in [1,N]} w_i x_i^2. \tag{3.30}$$

As we have only quadratic terms, the minimizer of the surrogate objective can be found by,

$$\mathbf{x}^{[n+1]} = \mathbf{W}\mathbf{D}^T(\mathbf{D}\mathbf{W}\mathbf{D}^T + \lambda\mathbf{I})^{-1}\mathbf{y}, \tag{3.31}$$

where $\mathbf{W} = \mathrm{diag}(\{w_i\}_{i \in [1,N]})$. To calculate $\mathbf{x}^{[n+1]}$ we need to invert a large matrix, which is not computationally possible for a large size problem. Similar to reweighted ℓ_1 approach, one can majorize the approximation error with a decoupled quadratic term and minimize the new majorizing function, which is equivalent to adaptively scaling each component of the Landweber update $\mathbf{a}$ (3.16), see [AD05, EMSZ07].

3.4.4 Other Sparse Approximation Methods

In the convex relaxed sparse approximation using iterative thresholding, it was mentioned that the unconstrained, but non-differentiable, optimization problem can be reformulated as the constrained differentiable problem (3.18). The new formulation is favorable to be solved using a quadratic programming and interior point method [CDS98]. Recently the interior point method has also been used directly to solve ℓ_1 regularized sparse approximation problem [KKL$^+$07]. Kim *et al.* [KKL$^+$07] used the primal logarithmic barrier method to solve the following equivalent problem,

$$\min_{\mathbf{x}} \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \sum_{i \in \mathcal{I}} u_i, \text{ s. t. } \forall i \in \mathcal{I} - u_i \leq \mathbf{x}_i \leq u_i, \tag{3.32}$$

using truncated Newton's method. The method uses an equality found using the dual form of (3.32) to simplify the problem and find an ϵ-suboptimal solution, where ϵ is the target duality gap. This technique can also be extended to the medium to large scale problems by solving the Newton system approximately.

Most of the methods reviewed in this chapter are based on (sub-)gradient or (sub)gradient-projection. Another fast method in this class is TwIST [BF07] in which the coefficient vector at the previous iteration $\mathbf{x}^{[n-1]}$ is also involved to calculate the new coefficient vector $\mathbf{x}^{[n+1]}$ in order to accelerate the convergence of the method. Recently another method was proposed by Wright *et al.* [WNF09] which adaptively changes the step size of the gradient step and can handle different separable sparsity measures.

Most of the sparse approximation methods based on the gradient-projection technique converge very slowly if λ is small. Such a small λ is interesting when the approximation error has to be small. In this case one can adaptively change λ, by starting from a large value, and accelerate the gradient projection method, see [DFL08]. The gradient projection technique can be applied to solve the LASSO problem (2.9) [vBF08]. Van den Berg *et al.* showed a relation between Basis Pursuit (BP), LASSO and Basis Pursuit DeNoising (BPDN)[9] problems, i.e. by choosing correct parameters, the problems share the solutions. This fact can help us to solve these problems using a gradient projection method, if the relation between the parameters are known. A method for solving such problems, by iteratively turning them to LASSO problems with different τ, has been presented in [vBF08].

As sparse approximation is formulated as an optimization problem, various optimization techniques have been applied to solve it. This chapter only covered the most often used algorithms and the algorithms which will later be used in this thesis. In the next section, a specific type of constrained sparse approximation problem is explored in which we are interested in finding a quantized value sparse approximation.

3.5 Quantized Sparse Approximations

In many applications of sparse coding we should store or send the coefficients in a quantized form. For example in the sparse audio [DD06], image [FVFK04, FV01] and video [AMN+99] coding we need to quantize the magnitude of the coefficients before (entropy) coding. In

[9] Here, it is also called convex relaxed sparse approximation.

contrast with the orthogonal transforms, the quantization error can be compensated by other non-zero coefficients. This has been demonstrated in [GVT98], followed by showing how an overcomplete representation can become consistant (see Chapter 1). The fact that the quantization error can be compensated using other coefficients, motivated the authors of [DZ03] to introduce a new greedy algorithm for quantized value sparse approximation, called In-loop Quantized MP. Their method has an extra step in the MP method, which is a quantization of the coefficients in each step, therefore it is called "in-loop quantized" MP. The quantization error, caused by the quantization of the coefficients, might be compensated in the following MP iterations. A similar technique can be applied to some of other iterative sparse approximation methods. A new quantized sparse approximation method, which is also published in [YBD07], based on the iterative hard thresholding, is introduced in the next subsection.

3.5.1 Quantized Sparse Approximation with the Majorization Method

In this section the problem of quantized sparse approximations will be considered and it will be shown that the problem can be solved using the majorization minimization method. The quantized version of (2.12) can be represented as,

$$\begin{aligned} &\min_{\mathbf{z}} \phi(\mathbf{z}) \\ &\phi(\mathbf{z}) = \|\mathbf{y}-\mathbf{D}\mathbf{z}\|_2^2 + \lambda \mathcal{J}(\mathbf{z}), \end{aligned} \tag{3.33}$$

where the sparsity measure $\mathcal{J}(.)$ is here selected to be ℓ_0 and $\mathbf{z}$ is a quantized value vector with the desired uniform quantizer, with larger zero bin. Let δ_0 and δ_1 be the zero and the non-zero bin sizes respectively. Therefore each component of $\mathbf{z}$ should be in $\mathcal{Z}$, which is defined as follows,

$$\mathcal{Z} = \{q_k\}_{k\in\mathbb{Z}} \ : \ q_k = \begin{cases} \frac{1}{2}(\delta_0 + k\delta_1) & k > 0 \\ 0 & k = 0 \\ \frac{1}{2}(-\delta_0 + k\delta_1) & k < 0 \end{cases} \tag{3.34}$$

In practice there exists an upper-bound for the magnitude of k which is related to the resolution of the quantizer L. The optimization problem (3.33) should be solved subject to $\mathbf{z}_i \in \mathcal{Z}$ for all $i \in [1, N]$. Iterative hard thresholding, see Section 3.4.2.2, can be used in the quantized domain. By adding the quantized version of the function defined in (3.13), $\pi_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^{\ddagger}) := c||\mathbf{z} - \mathbf{z}^{\ddagger}||_2^2 - ||\mathbf{D}\mathbf{z} - \mathbf{D}\mathbf{z}^{\ddagger}||_2^2$, to the objective of (3.33) and assuming that $\mathbf{z}^{\ddagger} = \mathbf{z}^{[n]}$, the following

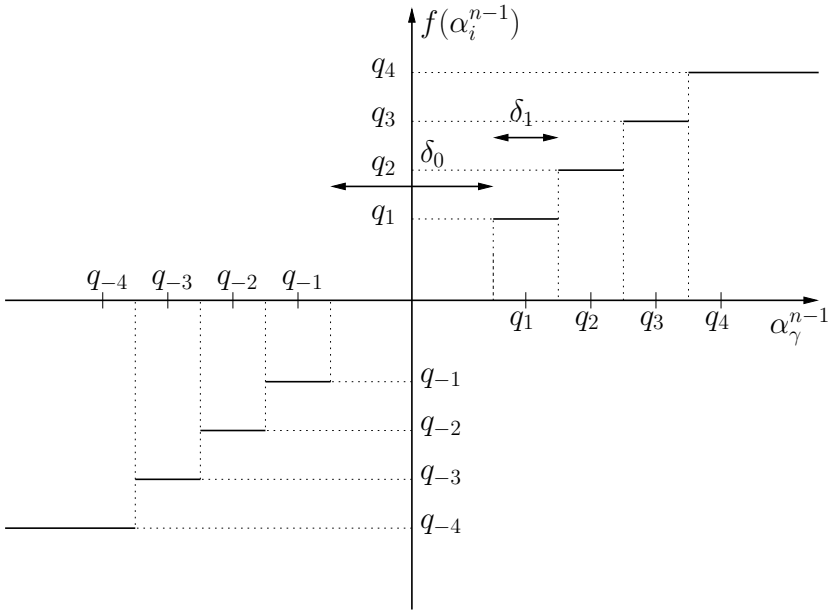

Figure 3.1: *9 level on-center QShrinker*

surrogate functions should be minimized at each step:

$$\{\psi(\mathbf{z},\mathbf{z}^{[n]})\}_i \propto cz_i^2 - 2z_i\left\{\mathbf{D}^T\left(\mathbf{y}-\mathbf{D}\mathbf{z}^{[n]}\right)+c\mathbf{z}^{[n]}\right\}_i + \lambda f(z_i), \tag{3.35}$$

where $f(.)$ was defined in (3.19). We are looking for the optimum value of $\psi(\mathbf{z},\mathbf{z}^{[n]})$ in the quantized value domain. Let $\mathbf{a}=\frac{1}{c}\left(\mathbf{D}^T\left(\mathbf{y}-\mathbf{D}\mathbf{z}^{[n]}\right)+c\mathbf{z}^{[n]}\right)$ be the Landweber update, which is the quantized version of what was defined in (3.16). The value of $\psi(\mathbf{z},\mathbf{z}^{[n]})$ can be evaluated at each z_i,

$$\{\psi(\mathbf{z},\mathbf{z}^{[n]})\}_i \propto \begin{cases} c(z_i-a_i)^2 - ca_i^2 + \lambda & z_i = q_{k:k\neq 0} \neq 0 \\ 0 & z_i = q_0 = 0 \end{cases} \tag{3.36}$$

where q_k is the k^{th} quantization level as defined in (3.34) ($k \in Z, -\lfloor L/2\rfloor+1 \leq k \leq \lfloor L/2\rfloor$ for an L level quantizer). Minimizing (3.36) is not difficult and we can find the minimizer by checking q_k's in the neighborhood of the current value of a_i. Note that the optimizer of $\{\psi(\mathbf{z},\mathbf{z}^{[n]})\}_i$ changes by λ, when a_i is close to zero. Therefore we can have different zero and adjacent bins than the quantizer proposed in (3.34). To have an in-loop operator similar to $\mathcal{Z}$, we can choose an appropriate λ, by using equation (3.37), see Fig. 3.1.

$$\lambda = (\delta_0/2)^2 - (\delta_1/2)^2 \tag{3.37}$$

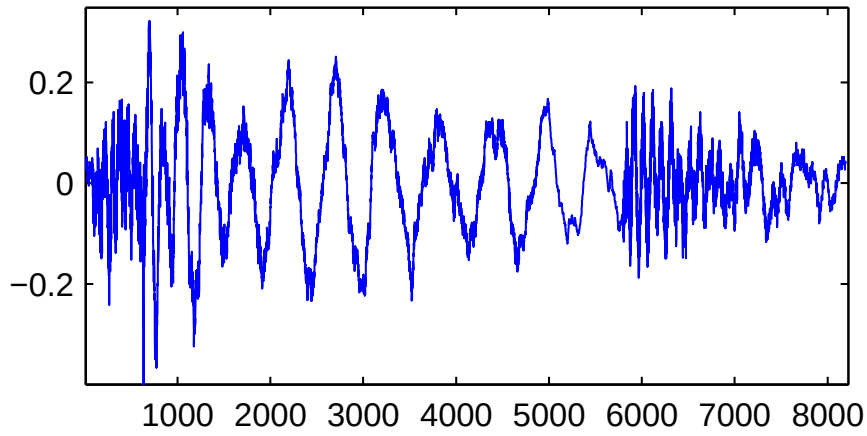

Figure 3.2: *Input audio signal*

Therefore the shrinking function changes to a simple uniform quantizer $f(.) = Q_{\mathcal{Z}}(.)$, where $\mathcal{Z}$ presents the centers of quantization bins of a uniform quantizer. As the problem has many local minima, the algorithm converges to different fixed points by starting with different initialization. Increasing the number of quantization levels, increasing the number of local minima. To improve performance, a relaxation strategy, which has previously been used in [Ela06], can be used for the iterative thresholding. Instead of updating the current coefficients with the proposed threshold, we choose a relaxation factor μ and update the current coefficients by,

$$x_i^{[n+1]} = (1-\mu)x_i^{[n]} + \mu f(\alpha_i), \tag{3.38}$$

where $0 < \mu \leq 1$. Note that the update x_i is no longer quantized. It is straightforward to show that the fixed points of both methods are similar. After the algorithm converges, all x_is have quantized values. Quantized Iterative Thresholding (QIT) can be summarize as calculating $\mathbf{a}$ and then using the operator $f(.)$. Because the values of the coefficients are quantized, the algorithm terminates when $\mathbf{x}^{[n+1]}$ is *exactly* equal to $\mathbf{x}^{[n]}$.

3.6 Simulations

A segment of pop music sampled at 32kHz was chosen here as a test signal (Figure 3.2). A 4 times overcomplete MDCT dictionary $\mathbf{D} \in \mathbb{R}^{1024\times 4096}$ (overcomplete in the frequency domain) was used. All simulations were started with $x^{[0]} = 0$. We fixed quantization levels and used a

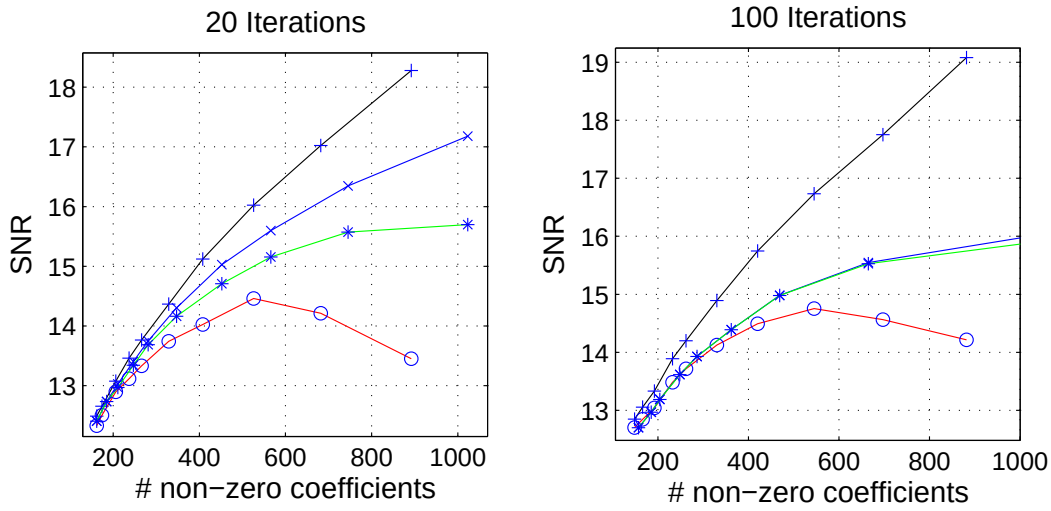

Figure 3.3: *For two different numbers of iterations (20 and 100) output SNR's are shown in four different cases (IT (+), QIT (x), quantized QIT (*), quantized IT (o))*

uniform dead-zone quantizer with the following zero bin to non-zero bin ratio,

$$\zeta = \frac{\delta_0}{\delta_1} \tag{3.39}$$

By changing ζ, the results of the algorithm will have a varying number of non-zero coefficients (it should be noted that this convention is not just for QIT. It is also used for IT, where the zero bin is the thresholding parameter. So we can compare equivalent coefficients quantized with QIT for a specific number of non-zero coefficients). A four bit quantizer (16 levels) was selected to quantize each coefficient. Simulations were run for 20 and 100 iterations to show the convergence of the algorithm. The results are shown in Figure 3.3. The graph with plus symbols is iterative hard thresholding and the results achieved when quantizing this solution are shown with circles. QIT and its quantized output are shown with cross and star symbols. Note that due to the relaxation approach used, the output of QIT is not automatically quantized. The horizontal axis shows the number of non-zero coefficients. We can see that for different numbers of non-zero coefficients, IT gives better SNR than QIT. However after quantization of the coefficients, the SNR of the decoded quantized coefficients of QIT is better than quantized IT. We also see that with more iterations, QIT and its quantized output get closer to each other, which shows that the algorithm is converging to a quantized solution.

Another observation to be made here is that the SNR starts to decrease when we use a large number of non-zero coefficients. This is caused by the fact that the optimal quantizer is changed

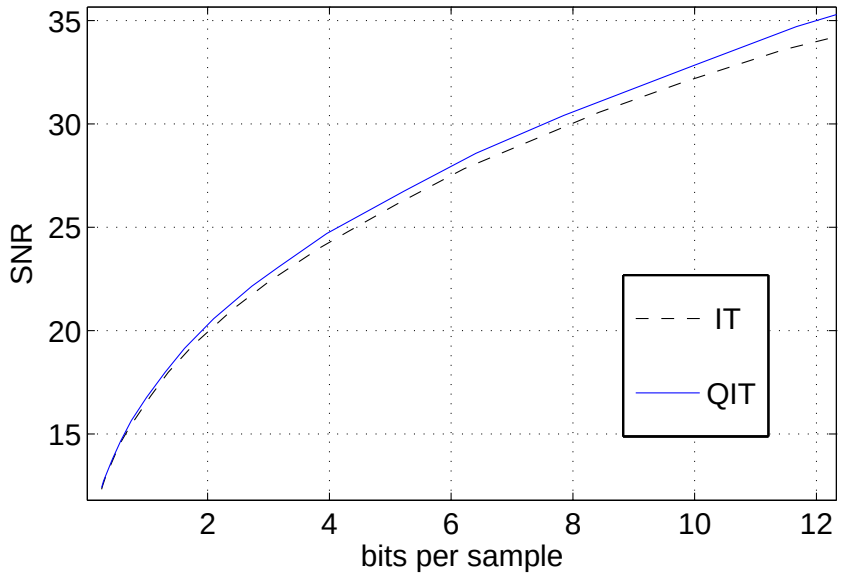

Figure 3.4: *Operating R-D curves for QIT (upper) and IT (lower)*

by changing the sparsity of the approximation. To show the benefit of using QIT, we need to show the operating rate-distortion (R-D) curve by computing the convex hull for different bit budgets. The audio sample used in the previous experiment is here used for coding with 4 to 9 bit quantizers. The operational R-D is shown in Figure 3.4. The graph shows that we have 0.2 dB SNR improvement for 1 bit/sample and up to 1 dB improvement for 12 bits/sample.

3.7 Summary

This chapter reviewed some of the sparse approximation methods. It was started by classifying the algorithms and presenting two important classes of the algorithms which are greedy and relaxed methods. The greedy methods, which are fast and suitable for large scale sparse approximation problems, is based on iteratively adding some atoms to the set of currently selected atoms and updating the coefficient values. These methods are easily implemented and have been guaranteed to have acceptable performances. On the other hand, the relaxed sparse approximation methods are mostly based on gradient descent, quadratic programming and interior point methods. If the sparse coding problem is convex, the analysis of the algorithm and its result are easier. The convex relaxed sparse coding methods have often been used in Compressed Sensing.

In the second part of this chapter the quantized sparse approximation problem was introduced and a new algorithm to solve it was proposed (first published in [YBD07]). Although this chapter does not address all the sparse approximation methods, it prepares the mathematical background and the algorithms are necessary to know for reading the rest of the thesis.

Part II

Dictionary Selection

Chapter 4
Dictionary Learning Formulation and State of the Art Algorithms

4.1 Introduction

In the first part of thesis, the sparse coding problem was formulated and it was assumed that the generative model, which was represented using a "dictionary", is given. This assumption is reasonable when the signals are generated synthetically or the dictionary is generated independent of the given signal. The first case is often observed, for example, in Synthetic Aperture Radar (SAR) imaging [VCFW08, PSZ08] and the latter is used in Compressed Sensing[1] [CRT06a, Don06]. Otherwise one needs to *a priori* select a dictionary. A suitable dictionary for a class of signals has to make sparse coding possible for the given signals. The optimal dictionary, in terms of the sparsest coding of the given class of signals, depends on an acceptable noise level in the model (2.5). As an example, the dictionary, which is suitable for a low noise sparse coding (in the high bit-rate sparse coding applications), is not optimal for a high noise sparse coding application [RRD08a]. This fact demonstrates the importance of dictionary selection methods.

Another parameter of a dictionary, which has to be selected *a priori*, is the size of dictionary. Often where the dictionary is not given, the size of the generative model is also unknown. On the other hand, when the dictionary size tends to infinity, the dictionary can include all the signals of the interest and the sparse coding is thus presenting the index of the related atom and the corresponding coefficient. It is clear that the sparse coding is not tractable and the coding cost of specifying the non-zero coefficient tends to infinity, by tending the dictionary size to infinity. We are always interested in a reasonable overcomplete dictionary size. A method to find the optimum dictionary size is thus of interest. The optimum dictionary size depends on the noise level in the model too. A framework for a minimum size dictionary learning will be presented in Chapter 6.

[1] The dictionary is called sensing matrix in CS, which is often generated randomly.

Sometimes the information about the signals of interest is given by a parametric function. In this case we assume that each atom is approximately represented by a set of parameters. The dictionary design problem is now formulated as how best to find the parameters. These parameters should be selected such that the class of signals has sparse representation/approximation using the parametric dictionary. Because the class of signals is not explicitly given, one can introduce an objective which should be optimized to increase the success rates of the exact sparse recovery and the sparsity of the signal approximations. A framework for parametric dictionary design is presented in Chapter 8. The proposed optimization problem is non-convex and difficult to solve. A practical algorithm to solve the problem approximately is also presented there.

The dictionary learning problem is formulated here while the formulation for a parametric dictionary design is postponed to be presented in Chapter 8. Numerous methods have been introduced to approximately solve the dictionary learning problem. This chapter can not explore all dictionary learning methods in detail and only tries to briefly explain the methods which have often been used.

4.2 Dictionary learning formulation

The dictionary learning problem is a kind of system identification problem. We here are interested in a linear discrete model which lies in a finite dimensional space. A major difference with the traditional system identification is that the input signals, which here is the coefficient vectors, are not given, i.e it is a blind system identification. Instead, an *a priori* model for the input signals is given, which promotes the sparsity of coefficient vectors. This makes the problem very difficult in general and the system identification techniques are often not applicable. One can also show that the complexity of the dictionary learning problem is at least in the order of sparse approximation problem, which was mentioned to be NP-hard in general.

In the dictionary learning problem, some constraints are often induced over the dictionary, which makes the dictionary learning problem *well-defined*. Some extra constraints have been induced to promote an extra structure, for example shift-invariance, for the dictionary. A different kind of constraint is induced to the dictionary in Chapter 7, to facilitate the learning process using less training samples. This framework can also be used to find a dictionary with a fast implementation, i.e. fast matrix-vector multiplication. We call a dictionary learning problem

"minimally constrained", if there only exists a constraint upon column or Frobenius norms of dictionaries. These constraints exclude trivial solutions of the dictionary learning problem. The Frobenius norm constraint not only makes the problem well defined, but also provides a local competition among the atoms, which provides a dictionary size reduction in practice. This is a motivation for the parsimonious dictionary learning in Chapter 6, by applying an extra sparsity penalty over the number of atoms.

In the dictionary learning we respectively use the sparse coding formulations (2.3) and (2.12), or their matrix forms, for sparse representations and approximations. A set of training signals $\mathscr{Y} = \{\mathbf{y}_i\}_{i\in\mathcal{L}}$, where $\mathcal{L}$ is the index set and $|\mathcal{L}| = L$, is given. $\mathscr{Y}$ should be large enough to represent the given class of signals. Let $\mathcal{D}$ be an admissible set of dictionaries. Dictionary learning for sparse representation, in general, is formulated as minimizing the following optimization problem,

$$\mathbf{D}^* = \arg\min_{\mathbf{D}\in\mathcal{D}}\{\min_{\mathbf{X}} \mathcal{J}(\mathbf{X}) \ \text{s.t.}\ \mathbf{Y} = \mathbf{D}\mathbf{X}\}, \tag{4.1}$$

where $\mathbf{Y} \in \mathbb{R}^{d\times L}$ is the matrix generated using $\mathbf{y}_i \in \mathscr{Y}$ as the columns. For a fixed (d, N, L), (4.1) is a non-convex optimization problem by selecting any sparsity measure $\mathcal{J}(.)$ and admissible set $\mathcal{D}$. There is no easy optimization method to exactly solve (4.1). Almost all the algorithms, even though there are few dictionary learning methods for sparse representations, can often find a local minimum of (4.1)[2]. This problem is much more difficult than the dictionary learning for sparse approximation, which will be explained latter. The difficulty is caused by the fact that the objective depends on one parameter and in a standard block-relaxed optimization framework, the objective can not be refined, while $\mathbf{X}$ is fixed. Although there exist some dictionary learning methods for sparse representations, see for example [Plu07a, PO06], those will not be covered in this thesis.

When the measured signals $\mathbf{y}_i$ in $\mathbf{Y}$ are noisy or they do not follow the linear generative model $\mathbf{Y} = \mathbf{D}\mathbf{X}$, the mismatch can be modeled as an additive noise. The dictionary learning problem in this setting is often defined as the following optimization problem,

$$\mathbf{D}^* = \arg\min_{\mathbf{D}\in\mathcal{D}}\{\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\mathcal{J}(\mathbf{X})\} \tag{4.2}$$

Although this problem is much easier than (4.1), it can not exactly be solved using standard

[2]There are some sampling methods which are proposed to find the global minimum of a non-convex objective, e.g. MCMC methods like Gibbs Sampler. These methods are often very computationally expensive and are tractable for small size problems.

optimization methods and typically the block-relaxation technique has been applied to find an approximate solution. The main difference between dictionary learning methods is how to select the blocks in a block-relaxed setting. The dictionary learning methods for sparse approximations are surveyed in Section 4.3. A novel method will be introduced in Chapter 5, which is very flexible in inducing different constraints over the dictionaries. The convergence of this method is guaranteed using an appropriate convex sparsity measure $\mathcal{J}(.)$ and convex admissible set $\mathcal{D}$.

4.3 Dictionary learning for sparse approximations

The dictionary learning methods for sparse approximations, which is for simplicity called "dictionary learning methods" here, are explored in this section. The methods often start with some initial dictionary and find the sparse approximations of a set of training signals, while keeping the dictionary fixed. It is followed by a second step in which the sparse coefficients are kept fixed and the dictionary is optimized. This alternating minimization method continues for a certain number of iterations or until a desired approximation error is reached. It was mentioned that the alternating minimization technique can be useful where the objective is based on both parameters, $\mathbf{X}$ and $\mathbf{D}$. As an example the objective of (2.3) is independent of $\mathbf{D}$ and can not therefore be optimized in this framework. Most of the dictionary learning methods are based on minimizing (4.2) in a block-relaxed framework using different sparsity measures and different parameter blocks [YBD09]. The sparsity measure $\mathcal{J}(.)$ and the parameter blocks are high-lighted when a dictionary learning algorithm is being explained in the rest of this section.

A dictionary learning problem is formulated for the first time[3] by Olshausen *et al.* in [OF97]. They modeled the strategy employed by V1 in the human vision system in a probabilistic framework. A prior distribution is being induced on the coefficient vectors in this framework. A probability distribution is chosen to promote the sparsity of approximations. Let (2.5) be the signal generative model[4] and $\mathbf{n}$ be an instance of an i.i.d. normal, zero mean random vector N with the variance σ, i.e. $n_i = \mathcal{N}(0, \sigma)$. Let the probability density function (pdf) of the random vector Y be noted by $f_Y(\mathbf{y})$, where $\mathbf{y}$ is an instance of Y. We are interested to find $\mathbf{D}$ such that $f_Y(\mathbf{y}|\mathbf{D})$ is as close as possible to $f_Y(\mathbf{y})$, where Y is a random vector represents

[3] According to what we here call *dictionary learning for sparse approximations.*

[4] Often different notations have been used in the sparsity induced probabilistic framework. To preserve the uniformity of thesis and preventing possible confusion, a similar notation to what has been used in Chapter 2 is chosen here.

the proposed class of signals. Here the closeness is measured using Kullback-Leibler divergence [CT91], which is equivalent to maximize the marginal log likelihood function. In this setting, the likelihood function is $f_Y(\mathbf{y}|\mathbf{D})$, which has been abbreviated by $\mathcal{L}(\mathbf{y}|\mathbf{D})$ here.

The dictionary learning in a log likelihood maximization framework is introduced in the next subsection. The derivation of the learning formula is easy and flexible such that it can be adapted to a structured dictionary learning problem, which will separately be reviewed in subsection 4.4.

4.3.1 Dictionary Learning using a Maximum Likelihood Estimator

An estimate for $\mathbf{D}$ can be found by maximizing the marginal likelihood function [OF97, OF96, LS00]. The first assumption, to make the problem tractable, is that the training samples are drawn independently and therefore,

$$\mathcal{L}(\mathbf{y}|\mathbf{D}) = \prod_{i \in \mathcal{I}} \mathcal{L}(\mathbf{y}_i|\mathbf{D}). \tag{4.3}$$

The likelihood function, for each training sample, is a conditional pdf of the coefficient vectors $\mathbf{x}$ and can be represented by $\mathcal{L}(\mathbf{y}_i|\mathbf{D}, \mathbf{x})$. Now for each $\mathbf{y}_i$, the likelihood function can be calculated as follows,

$$\mathcal{L}(\mathbf{y}_i|\mathbf{D}) = \int \mathcal{L}(\mathbf{y}_i|\mathbf{D}, \mathbf{x}) f_X(\mathbf{x}) d\mathbf{x}. \tag{4.4}$$

The integral is calculated over the instances of the random variable X. It was mentioned in Chapter 3 that a suitable distribution should be selected to promote the sparsity of representations. Cauchy [OF97] and Laplace [OF96, LS00] distributions have been chosen as the pdf of X, $f_X(\mathbf{x})$. The maximum log-likelihood[5] dictionary learning can be formulated as,

$$\mathbf{D}^* = \arg\max_{\mathbf{D}} \log \mathcal{L}(\mathbf{y}|\mathbf{D}) = \arg\max_{\mathbf{D}} \log \prod_{i \in \mathcal{I}} \int \mathcal{L}(\mathbf{y}_i|\mathbf{D}, \mathbf{x}) f_X(\mathbf{x}) d\mathbf{x}. \tag{4.5}$$

[5]The logarithm is a strictly increasing operator in the positive orthant. Therefore maximum log-likelihood and likelihood share the same solution set. Here it is preferred to maximize log-likelihood for its easier derivation of the update formula.

When N and X respectively follow the i.i.d normal and Laplace distributions, (4.5) can be rewritten by,

$$\mathbf{D}^* = \arg\max_{\mathbf{D}} \sum_{i\in\mathcal{I}} \log \int \exp\{\frac{-1}{2\sigma^2}\|\mathbf{D}\mathbf{x}-\mathbf{y}_i\|^2\}.\exp\{-\lambda\|\mathbf{x}\|_1\}d\mathbf{x}. \tag{4.6}$$

Unfortunately maximization of the log-likelihood is not easy. Different methods have been presented to approximately solve this problem. Olshausen *et al.* [OF97] approximate the volume under the surface, defined by the terms inside integral, by its maximum. The optimal dictionary can now be found by solving the following joint optimization problem,

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sum_{i\in\mathcal{I}} \min_{\mathbf{x}_i}\{\|\mathbf{D}\mathbf{x}_i-\mathbf{y}_i\|^2 + 2\sigma^2\lambda\mathcal{J}(\mathbf{x}_i)\}, \tag{4.7}$$

where $\mathcal{J}(.)$ is the sparsity promoting operator related to the prior distribution of the coefficients. The operator $\mathcal{J}(.)$ is respectively $\|.\|_1$ and $\sum_{1\le i\le N}\log(1+\{.\}_i^2)$ for the Laplace and Cauchy prior distributions. Olshausen *et al.* suggested in [OF97] to solve (4.7) by minimizing based on $\mathbf{x}$, while keeping $\mathbf{D}$ fixed, in the first step and then update $\mathbf{D}$ such that it reduces the objective using a gradient descent method and repeating this alternating minimization to converge to a solution. The dictionary update can be found using,

$$\mathbf{D}^{[n+1]} = \mathbf{D}^{[n]} - \eta\sum_{i\in\mathcal{I}}(\mathbf{D}^{[n]}\mathbf{x}_i-\mathbf{y}_i)\mathbf{x}_i^T, \tag{4.8}$$

where η is a suitable step size. If the norm of $\mathbf{D}$ is not constrained to be bounded in (4.7), the solution $\mathbf{D}^*$ tends to infinity. This is caused by what is called scale-ambiguity, $\forall(\alpha<1)\in\mathbb{R}^+$, if $(\mathbf{D}^*,\mathscr{X}^*)$ is a pair of optimal dictionary and the set of corresponding optimal non-zero coefficient vectors respectively, $(\frac{1}{\alpha}\mathbf{D}^*,\alpha\mathscr{X}^*)$ is a better solution for (4.7), where $\alpha\mathscr{X}^*$ indicate the set of $\{\alpha\mathbf{x}_i^*\}_{i\in\mathcal{I}}$ in which the better means that the objective of (4.7) is reduced by updating $(\mathbf{D}^*,\mathscr{X}^*)$ with $(\frac{1}{\alpha}\mathbf{D}^*,\alpha\mathscr{X}^*)$[6]. Olshausen *et al.* [OF97] used an extra atom renormalization step to resolve this problem. This modification actually constrains the dictionaries to stay in $\mathcal{D}=\{\mathbf{D}_{d\times N}:\|\mathbf{d}_i\|_2=c_C^{1/2}\}$. A disadvantage of this method is that the objective of (4.7) might increase after projecting to the admissible set $\mathcal{D}$, which is done by rescaling. Therefore there is no convergence guarantee for this algorithm.

[6] A solution of (4.7) in which $\|\mathbf{D}\|\to\infty$ and $\forall i:\|\mathbf{x}_i\|\to 0$ is sometimes called a "Degenerate Solution" [KMR+03].

Approximating the integral of (4.6) with the maximum of its integrand lets us solve ML problem easier, but it introduces an error into the result. A different approach to solve (4.6) is to approximate $\mathcal{L}(\mathbf{y}_i|\mathbf{D},\mathbf{x})f_X(\mathbf{x})$ by a Gaussian and using the Gaussian integral approximation [LS00, (A.2)], which is often called Laplace's method. The priori pdf for the coefficients is a non-smooth function and needs to be approximated by a two-times differentiable function to allows computation of the Hessian. Here the Laplace distribution is approximated by the following function:

$$f_X(\mathbf{x}) = \prod_{i\in\mathcal{I}} \exp(-\lambda|x_i|) \approx c \prod_{i\in\mathcal{I}} \cosh^{-\lambda/\beta}(\beta x_i) \tag{4.9}$$

Where c and β are some constants. The surrogate function more accurately approximates $f_X(\mathbf{x})$ with large β and is smooth around zero, which is the discontinuity of Laplace pdf. These simplifications allow us to calculate the gradient of objective in (4.5) and maximize it using a gradient ascent method. In this framework the gradient with respect to $\mathbf{D}$ can be found by,

$$\frac{\partial}{\partial \mathbf{D}}\mathcal{L}(\mathbf{y}_i|\mathbf{D}) = -\mathbf{D}\Big(\frac{\partial \log f_X(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}^{[n]}} \mathbf{x}^{[n]^T} + \mathbf{I}\Big). \tag{4.10}$$

where $\mathbf{x}^{[n]}$ is the current sparse representation of $\mathbf{y}_i$ and $\partial \log f_X(\mathbf{x})/\partial \mathbf{x} = \{\partial \log f_X(x_i)/\partial x_i\}$ is calculated using the approximation (4.9). (4.10) is the gradient based on one training sample and the dictionary is updated only with respect to that sample. If the model and distributions comply with the training samples, the algorithm converges to a solution. In the case of model mismatch or noisy training samples, the algorithm might diverge.

In practice a step size should also be selected to update $\mathbf{D}$ in the gradient direction. A simple fixed step size is used in [LS00]. This can be pessimistic if the selected Gaussian approximation does not accurately approximate the volume under $f_X(\mathbf{x})$. To improve the accuracy of the approximation and accelerate the algorithm's convergence, an adaptive step size can be used [LS00], where the accuracy of approximations are demonstrated in [LS00, Fig. 4]. The new scheme selects a step size that reduces the posterior log-likelihood objective for a fixed drop, which is found by optimally fitting a Gaussian distribution to a Laplace distribution. The reader is referred to [LS00] for more details on the algorithm and its derivation.

The methods explained to maximize the (log-) likelihood function (4.3) are based on a technique called stochastic gradient descent [KY03]. An alternative to this is to use a Monte Carlo

(MC) method [OM00, BD07]. An advantage of using MC is that the estimation is unbiased in contrast to the former methods. The MC method can be applied using the importance and Gibbs sampler in a Markov Chain setting, see for example [Blu06] and references therein.

The ML framework has been used for dictionary learning by many researchers and promising results have been reported [SL06, BD06, Lew02, OF97, OF96]. An important issue with most of the ML based methods is scale ambiguity, which is often compensated by projecting back onto a given admissible sets. Although it can resolve the problem in most practical applications, there is no mathematical analysis for the algorithms. It is also not clear which objective is actually minimized using the proposed algorithms. Another possible way to resolve this issue is to formulate the dictionary problem as a Maximum A Posteriori (MAP) problem, where a prior distribution is assumed for the dictionary. The dictionary thus can not get arbitrary large in this setting. This type of dictionary learning will be introduced in the next subsection in more detail.

4.3.2 Dictionary Learning using a Maximum A Posteriori Estimator

In dictionary learning using ML estimation, the coefficients follow a prior distribution and the dictionary is assumed to be deterministic. The constraint on the dictionary, which is presented by an admissible set, is thus deterministic. An alternative is to apply such a constraint on the dictionary by introducing a prior distribution on the dictionary. Now the dictionary learning problem is defined completely in the stochastic domain and can be estimated using, for example, a MAP estimator. In the setting used in Subsection 4.3.1, the posterior $f_{D,\mathbf{x}|Y}(\mathbf{D}, \mathbf{X}|\mathbf{y})$, where $\mathbf{X} \in \mathbf{R}^{N\times L}$ is an instance of the random matrix X in which each column is independently generated using instances of $\mathbf{x}$, should be maximized. Let $f_{D,\mathbf{x}|Y}(\mathbf{D}, \mathbf{X}|\mathbf{y})$ be abbreviated by $\mathcal{P}(\mathbf{D}, \mathbf{X}|\mathbf{y})$. The posterior pdf can be reformulated using Bayes rule as follows,

$$\mathcal{P}(\mathbf{D}, \mathbf{X}|\mathbf{y}) \approx \mathcal{L}(\mathbf{y}|\mathbf{D}, \mathbf{X}) f_D(\mathbf{D}) f_X(\mathbf{X}). \tag{4.11}$$

If $f_D(\mathbf{D})$ is flat, i.e. there is no preference for any $\mathbf{D}$, the estimation would be the same as ML estimation by approximating the marginal likelihood with its maximum [OF97]. Otherwise MAP estimates a dictionary which is most probable in the posterior distribution. It has similarities with constraining the dictionary to lie in an admissible set. $f_D(\mathbf{D})$ is corresponded to an admissible set $\mathcal{D}$, if it has a uniform distribution in the admissible set and zero outside, i.e. it is flat over a compact set (otherwise $f_D(\mathbf{D})$ is not well-defined). The dictionary learning in the

MAP framework can be extended by assuming that $f_D(\mathbf{D})$ has a more realistic pdf by changing the uniform distribution to another distribution which has larger values for more appealing dictionaries, for example being sparse to have less complex implementation[7]. Although it is no longer a MAP estimation in the new setting, it can be treated similarly. Here the dictionary update formula is only derived for the uniform $f_D(\mathbf{D})$ over the admissible sets $\mathcal{D}$ [KMR+03]. In this setting, the unit column and Frobenius norms dictionary sets have been used as $\mathcal{D}$ and the update formulas have been derived. In this framework, the dictionary learning can be reformulated as,

$$\mathbf{D}^* = \text{argmax}_{\mathbf{D}} \max_{\mathbf{X}} \log \mathcal{P}(\mathbf{D}, \mathbf{X}|\mathbf{y}) = \text{argmax}_{\mathbf{D}\in\mathcal{D}} \max_{\mathbf{X}} \log \prod_{i\in\mathcal{I}} \mathcal{L}(\mathbf{y}_i|\mathbf{D}, \mathbf{x}_i) f_X(\mathbf{x}_i). \tag{4.12}$$

The term $f_D(\mathbf{D})$ is cancelled out because of the uniformity of the distribution across $\mathcal{D}$. This formulation for the Gaussian noise and a super-Gaussian coefficients with $f_X(\mathbf{x}) = \prod_{i\in\mathcal{I}} e^{-\lambda|x_i|^p}$ and $p : 0 \leq p \leq 1$ can be rewritten as follows,

$$\begin{aligned} \mathbf{D}^* &= \text{argmin}_{\mathbf{D}\in\mathcal{D}} \min_{\mathbf{X}} \Phi(\mathbf{D}, \mathbf{X}) \\ \Phi(\mathbf{D}, \mathbf{X}) &= \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + 2\sigma^2\lambda\mathcal{J}_p(\mathbf{X}), \end{aligned} \tag{4.13}$$

where $\mathcal{J}_p(.)$ was defined before in (2.18). This optimization problem is non-convex and difficult to solve. The method introduced to solve this problem is based on alternating minimization [KMR+03]. This method is a particular case of the block-relaxed optimization technique, which is also used in next section. The optimization is done in two steps in this framework, where one parameter set is optimized at each step and the other is kept fixed. When $\mathbf{D}$ is fixed, (4.13) is a standard sparse approximation and can be solved using Iterative Reweighted ℓ_2 method [GR97] which was presented in subsection 3.4.3.2. The other step, which is the dictionary update, has been done using a gradient projection technique. This technique is very powerful as a constrained optimization method as long as the objective is differentiable. This is an iterative method which has two distinct parts at each iteration, I) Dictionary update in the

[7] As long as the learned dictionary does not have any structure for a fast matrix-vector multiplication, sparser dictionary has less number of element-wise multiplications at each implementation, which reduces the computation cost.

negative gradient direction. II) Projecting the new dictionary onto the admissible set (or onto the tangent space of the admissible set). Kreutz Delgado *et al.* [KMR$^+$03] chose the projection onto the tangent space[8]. In this framework, each iteration of the update step can be formulated as,

$$\mathbf{D}^{[n+1]} = \mathbf{D}^{[n]} - \gamma \, \mathcal{P}_{\mathcal{T}_\mathcal{D}} \left(\left. \frac{\partial \Phi}{\partial \mathbf{D}} \right|_{\mathbf{D}^{[n]}} \right), \tag{4.14}$$

where $\mathcal{P}_{\mathcal{T}_\mathcal{D}}$ is the orthogonal projection onto the tangent space $\mathcal{T}_\mathcal{D}$ of the admissible set $\mathcal{D}$. Now the update formula for the unit column and the unit Frobenius norm admissible sets can be driven using (4.14). The sketches of the derivation of formulas will be given here. For a more detailed explanation an interested reader can refer to [KMR$^+$03, Appendices A and B].

The first step is to compute $\partial \mathbf{\Phi} / \partial \mathbf{D}$, while $\mathbf{X}$ is kept fixed, which can be done using differential analysis of matrix value functions, see for example [Dat09], as follows,

$$\left. \frac{\partial \Phi}{\partial \mathbf{D}} \right|_{\mathbf{D}^{[n]}} = -\mathbf{E}\mathbf{X}, \tag{4.15}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{D}^{[n]}\mathbf{X} = \sum_{i \in \mathcal{I}} \mathbf{y}_i - \mathbf{D}^{[n]}\mathbf{x}_i$ is the approximation error. The next step is projection onto the tangent space of the admissible set at the current dictionary $\mathbf{D}^{[n]}$, which is shown in the following for each admissible set.

1. *Unit Frobenius Norm Dictionaries:* Let $\mathcal{D} = \{\forall \mathbf{D} \in \mathbb{R}^{d \times N} : \|\mathbf{D}\|_F = 1\}$. The projection of a matrix $\mathbf{Q} \in \mathbb{R}^{d \times N}$ onto the tangent space of $\mathcal{D}$ at $\mathbf{D}^{[n]}$, which was found for example in [KMR$^+$03], can be presented by the following operator,

$$\mathcal{P}_{\mathcal{T}_\mathcal{D}}(\mathbf{Q}) = \mathbf{Q} - \operatorname{tr}\{\mathbf{Q}^T \mathbf{D}^{[n]}\} \mathbf{D}^{[n]}. \tag{4.16}$$

   The dictionary update can thus be found using the following formula,

$$\mathbf{D}^{[n+1]} = \left(1 - \gamma \operatorname{tr}\{\mathbf{D}^{[n]^T} \mathbf{E}\mathbf{X}^T\}\right) \mathbf{D}^{[n]} + \gamma \mathbf{E}\mathbf{X}^T, \tag{4.17}$$

   where $\gamma > 0$ is the update step size which should be selected wisely to guarantee the convergence of the iterative algorithm.

2. *Unit Column Norm Dictionaries:* The admissible set is now defined by $\mathcal{D} = \{\forall \mathbf{d}_i \in \mathbb{R}^d : \|\mathbf{d}_i\|_2 = 1\}$. Here the constraint is column separable and the projection onto the

[8] They basically derived the formulation using a fixed-point continuation technique. It is also not difficult to derive the same formulation using gradient projection onto the tangent space.

tangent space of the unit atom norm, called $\mathcal{D}_i$ for the ith atom, can easily be found by the following operator,

$$\mathcal{P}_{\mathcal{T}_{\mathcal{D}_i}}(\mathbf{q}) = \left(\mathbf{I} - \mathbf{d}_i^{[n]}\mathbf{d}_i^{[n]^T}\right)\mathbf{q}. \tag{4.18}$$

The update of each atom $\mathbf{d}_i$ can thus be found using (4.14) and (4.18) with the following formula,

$$\mathbf{d}_i^{[n+1]} = \mathbf{d}_i^{[n]} + \gamma\left(\mathbf{I} - \mathbf{d}_i^{[n]}\mathbf{d}_i^{[n]^T}\right)\mathbf{E}\mathbf{x}_i^T. \tag{4.19}$$

This formula is an atom-wise update and therefore should be applied on each atom consecutively which slows down the update operation in practice, see for example [AEB06] and the simulations of Chapter 5.

The dictionary can be updated for a certain number of iterations using (4.17) or (4.19) and then switched to update the other parameter, $\mathbf{X}$. The overall convergence rate of the algorithm is depends on the update step size, where a larger step size accelerates the algorithm, it might make the algorithm unstable.

The optimization problem which is derived in a MAP estimation framework is the general form of the dictionary learning problem (4.2). Therefore one can initially start with solving (4.2), using a more efficient optimization technique. Such a minimization method will be introduced in Chapter 5. The new method shows faster convergence in practice and it is easier to implement.

Another important dictionary learning methods, which will be explored in the following, are in a deterministic framework. The Method of Optimal Directions (MOD) has been presented in [EAH99a], which has similarities with the dictionary learning methods based on ML. The MOD method will be introduced in the next subsection.

4.3.3 Method of Optimal Directions (MOD)

The MOD method is inspired by the Generalized Lloyd Algorithm (GLA), which has been used for designing VQ code books [GG91]. It is supposed to solve (4.2) using alternating minimization in this method. The sparse coefficients are firstly found using a sparse approximation method, for example MP [EAH99a], OMP [EAH99b] or FOCUSS (Iterative Reweighted ℓ_2) [ERK99], then the constrained optimization problem, with respect to $\mathbf{D}$, is temporarily

relaxed using $\mathcal{D} = \mathbb{R}^{d \times N}$. This can be formulated by the following optimization problem,

$$\mathbf{D}^{[n+1]} = \arg\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{[n]}\|_F^2, \tag{4.20}$$

where $\mathbf{X}^{[n]}$ is the sparse matrix found by the sparse approximation of $\mathbf{Y}$ using $\mathbf{D}^{[n]}$ as the dictionary. This is a standard convex optimization problem and can be solved using different techniques [BV04]. The objective is strictly convex, if $\mathbf{X}^{[n]}$ is full rank, and (4.20) has thus a unique solution:

$$\mathbf{D}^* = \mathbf{Y}\mathbf{X}^{[n]^\dagger} := \mathbf{Y}\mathbf{X}^{[n]^T}(\mathbf{X}^{[n]}\mathbf{X}^{[n]^T})^{-1}, \tag{4.21}$$

which is called the Moore-Penrose pseudoinverse [Moo20, Pen55, GVL96]. The next step is to project onto the admissible set $\mathcal{D}$. Here $\mathcal{D}$ is selected to be the unit column norm matrices. Therefore the columns of $\mathbf{D}^*$ should be re-normalized. If the norm of any column of $\mathbf{D}^*$ is zero, or very small, one can substitute that atom with a random vector to preserve the size of dictionary. Although no divergence in the algorithm is reported in [Eng00], the convergence analysis is challenging. A difficulty with implementing MOD occurs when the dictionary is close to being singular, which makes the calculation of the inverse of $\mathbf{X}^{[n]}\mathbf{X}^{[n]^T}$ practically intractable.

(4.20) can also be solved using the gradient descent method, which provides an update formula similar to what was found in the dictionary learing using ML method, Subsection 4.3.1, where the marginal likelihood is approximated by its maximum. Another method to solve (4.20) is to use a conjugate gradient method, which increases the convergence rate of the algorithm.

4.3.4 K-SVD Dictionary Learning

Sparse approximation of the signals can be seen as a generalization of the classification problem. Let each class be represented by a *class indicator*, which is simply a point in the vector space. The classification aims to assign one class indicator to each data point. This is equivalent to 1-sparse approximation of the signals, where the class indicators are the atoms. Therefore sparse approximation can be interpreted as an extention of the classification, letting each data point be indicated by a weighted superposition of k atoms for $k > 1$. In this setting, the dictionary learning can be interpreted as clustering, which is supposed to find the class indicators, when they are not available. A special case of the clustering problem is the k-means probelm, which uses the ℓ_2 norm as the distance measure and the mean of a cluster as its indicator. The

k-means problem is in general an NP-hard problem [Bru77, DFK$^+$04]. K-SVD algorithm is inspired from the k-means algorithm, which is also called GLA [LBG80].

K-SVD algorithm is introduced to solve (4.2), using $\mathcal{D} = \{\forall \mathbf{d}_i \in \mathbb{R}^d : \|\mathbf{d}_i\|_2 = 1\}$ as the admissible set and $\mathcal{J}(.) = \ell_0$. The quadratic term of (4.2) and ℓ_0 are column separable operators with respect to $\mathbf{X}$, which allows us to solve the sparse approximation using conventional sparse *vector* approximation methods. Here OMP has been used to approximately solve the sparse approximation step. The next step is to update the dictionary, which is done simultaneously with updating the non-zero coefficients. Aharon *et al.* [AEB06] simplified this operation using a block-relaxed optimization technique. Each block includes one atom and corresponding coefficients, which generate one row of $\mathbf{X}$. Now we have N such blocks of parameters and the optimization problem can be relaxed to only optimize based on each block of parameters. Such an optimization problem can be formulated as follows,

$$(\mathbf{d}_i^{[n+1]}, \mathbf{x}^{(i)^{[n+1]}}) = \arg \min_{(\mathbf{d}\in\mathcal{D}_i, \bar{\mathbf{x}}\in\mathbb{R}^N)} \|\mathbf{E}_{\mathcal{I}\setminus i} - \mathbf{d}\bar{\mathbf{x}}^T\|_F^2 + \lambda\|\bar{\mathbf{x}}\|_0, \tag{4.22}$$

where $\mathbf{x}^{(i)^{[n+1]}}$, $\mathbf{E}_{\mathcal{I}\setminus i}$ and $\mathcal{D}_i$ respectively are the updated i^{th} row of $\mathbf{X}$, the signal approximation error using all, but the i^{th}, atoms and corresponding coefficients at the n^{th} iteration, which can be found by $\mathbf{E}_{\mathcal{I}\setminus i} := \mathbf{Y} - \sum_{\mathcal{I}\setminus i} \mathbf{d}_i^{[n]} \tilde{\mathbf{x}}^{(i)^{[n+1]^T}}$[9] and the admissible set of the i^{th} atom $\mathcal{D}_i = \{\forall \mathbf{d} \in \mathbb{R}^d : \|\mathbf{d}\|_2 = 1\}$. The rank of matrix $\mathbf{d}\bar{\mathbf{x}}^T$ is one. Therefore the problem can be interpreted as finding a rank one matrix close to $\mathbf{E}_{\mathcal{I}\setminus i}$, where the $\bar{\mathbf{x}}$ is sparse.

There exist algorithms to find the closest rank one matrix to a matrix, which give us $\mathbf{d}$ and $\bar{\mathbf{x}}$. Let the singular value decomposition of $\mathbf{E}_{\mathcal{I}\setminus i}$ be,

$$\mathbf{E}_{\mathcal{I}\setminus i} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{4.23}$$

where $\mathbf{U} \in \mathbb{R}^{d\times d}$ is a unitary matrix, $\mathbf{V} \in \mathbb{R}^{L\times d}$ is a collection of d orthonormal vectors in $\mathbb{R}^L$, i.e. $\mathbf{V}^T\mathbf{V}$ is identity matrix, and $\mathbf{\Sigma} = \text{diag}\{\sigma_i\}_{1\le i\le d} \in \mathbb{R}^{d\times d}$. Also let the singular values σ_i are ordered based on the magnitudes such that $\forall i \in \mathbb{N}, 1 \le i < d : \sigma_i \ge \sigma_{i+1}$. Otherwise, it can easily be ordered by simultaneously swapping the columns of $\mathbf{U}$ and $\mathbf{V}$. In the norm space of matrices in $\mathbb{R}^{d\times L}$, equipped with the Frobenius norm, the closest rank one approximation of

[9] $\tilde{\mathbf{x}}^{(i)^{[n+1]}}$ is the i^{th} row of $\tilde{\mathbf{X}}^{[n+1]}$, which is the updated $\mathbf{X}^{[n]}$ using OMP in the $n + 1^{th}$ iteration. The coefficient matrix will also be updated in the dictionary update step to generate $\mathbf{X}^{[n+1]}$. The tilde symbol has been used to indicate the value of $\mathbf{X}$ in such a transient step.

$\mathbf{E}_{\mathcal{I}\setminus i}$ can be found by,

$$\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T, \tag{4.24}$$

where $\mathbf{v}_1$ and $\mathbf{u}_1$ are the first columns of $\mathbf{V}$ and $\mathbf{U}$ respectively. This approximation is unique if σ_1 is strictly grater than σ_2. This classical result can be used to solve (4.22) by choosing $\mathbf{d}_i^{[n+1]}$ and $\mathbf{x}^{(i)^{[n+1]}}$ as $\mathbf{u}_1$ and $\sigma_1 \mathbf{v}_1$ respectively. Unfortunately $\sigma_1 \mathbf{v}_1$ is not sparse in general. Aharon *et al.* in [AEB06] relaxed (4.22) by keeping fixed the sparsity penalty during optimization and by forcing the zero components to remain zero. Let $\omega^{(i)}$ be the index set of non-zero components of $\tilde{\mathbf{x}}^{(i)^{[n]}}$. To force $\mathbf{x}^{(i)^{[n+1]}}$ and $\tilde{\mathbf{x}}^{(i)^{[n+1]}}$ have a similar support, the problem of best rank one approximation can be solved with the input matrix $\mathbf{E}_{\mathcal{I}\setminus i}^{\omega^{(i)}}$ which is the shrunk version of $\mathbf{E}_{\mathcal{I}\setminus i}$ generated only using the columns indexed by $\omega^{(i)}$. Now we can use (4.24) to find $\mathbf{d}_i^{[n+1]}$ and the non-zero components of $\mathbf{x}^{(i)^{n+1}}$ indexed by $\omega^{(i)}$, which respectively are $\mathbf{u}_1$ and $\sigma \mathbf{v}_1$. The dictionary and the sparse approximation are updated when we iterate through all atoms. The updating atom is selected randomly or based on the atom indices in the dictionary.

The algorithm can be run for a certain number of alternating updates of $\mathbf{X}$ and $(\mathbf{D}, \mathbf{X})$. An issue with the stability analysis of the algorithm is that OMP is used to solve ℓ_0 sparse approximation, which always has some error. Therefore at some iterations, particularly when the solution becomes close to a fixed point, the objective might increase in some updates.

The SVD of $\mathbf{E}_{\mathcal{I}\setminus i}$ should be calculated for all i at each iterations. The K-SVD algorithm only needs the largest singular value and the corresponding singular vector, which can be calculated more efficiently [GVL96]. As an alternative, one can solve (4.22), after fixing the support of $\bar{\mathbf{x}}$, in an alternating minimization framework [RZE08]. The problem is not convex based on both $\mathbf{d}$ and $\bar{\mathbf{x}}$ and we only find a local minimum of the relaxed version of (4.22), by fixing support of $\bar{\mathbf{x}}$, which is shown that the solutions perform well in practice [**?**].

The K-SVD method is distinguished from previously mentioned methods mainly for the simultaneous updating the dictionary and the non-zero coefficients in the dictionary update step. It accelerates the convergence of the algorithm to a fixed point. Note that, as the dictionary learning problem is a non-convex problem, we are only looking for a "good" fixed point or local minimum. Aharon *et al.* showed that K-SVD also find reasonably good solutions. It is worth mentioning that, although K-SVD converges fast, each iteration of the algorithm is computationally expensive as we need to N-times calculate SVD at each single iteration.

4.3.5 Other Dictionary Learning Methods

Other dictionary learning methods are also based on minimizing an objective like the objective in (4.2), which has been derived in a deterministic or a stochastic framework using different optimization techniques. Some of the optimization techniques are explained here after specifying the objective and the admissible sets.

The optimization problem (4.2) has been considered in [LBRN07] which has a bounded column norm admissible set, defined by $\mathcal{D} = \{\forall \mathbf{d}_i \in \mathbb{R}^d : \|\mathbf{d}_i\|_2 = c\}$, where c is a constant. An advantage of such an admissible is that it is *convex* and, as long as $\mathcal{J}(.)$ is convex, the objective becomes a *bi-convex* problem. In other words, the objective is convex with respect to each parameter ($\mathbf{X}$ and $\mathbf{D}$), over a convex admissible set, while the other parameter is kept fixed. It allowes us to use variety of convex optimization techniques in a block-relaxed framework. This *bi-convex* objective is also used in the dictionary learning method with the majorization minimization method, which will be presented in Chapter 5. Lee *et al.* proposed an algorithm which finds the support of the coefficient matrix in one step of alternating minimization. The technique is based on relaxing the ℓ_1 penalty term by assuming a prior information about the sign of $x_{i,j}$, and updating the sign information based on the actual sign of the coefficients in each estimation of $\mathbf{X}$. Although it is an interesting algorithm, more investigations are necessary. As this chapter is about dictionary learning, the dictionary update method of [LBRN07] is explained in more detail. In the dictionary update step, which is a constrained convex optimization problem, the dual problem is solved by the Newton's method. The motivation is that the number of unknown parameters is reduced significantly in the dual space. The coefficient matrix $\mathbf{X}$ is fixed during the update. The optimization problem in this step can be rewritten as,

$$\mathbf{D}^{[n+1]} = \arg\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{[n]}\|_F^2. \tag{4.25}$$

This is almost the same as (4.20), but the solution is constrained to be in $\mathcal{D}$. Using Lagrangian multipliers [Roc70] $\Lambda = \mathrm{diag}\{\lambda_i\}_{i\in\mathcal{I}}$, where $\forall\lambda_i \geq 0$, the dual optimization problem [BV04] can be found by,

$$\begin{aligned} \mathbf{D}^{[n+1]} &= \arg_{\mathbf{D}} \max_{\Lambda} \min_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \Lambda) \\ \mathcal{L}(\mathbf{D}, \Lambda) &= \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{[n]}\|_F^2 + \mathrm{trace}\{\mathbf{D}^T\Lambda\mathbf{D} - c\Lambda\}. \end{aligned} \tag{4.26}$$

For any admissible Λ[10], $\mathcal{L}(\mathbf{D}, \Lambda)$ is strictly convex with respect to $\mathbf{D}$. The minimum of $\mathcal{L}(\mathbf{D}, \Lambda)$ can be found by letting the gradient being zero. (4.26) can now be simplified as follows,

$$\begin{aligned} \mathbf{D}^{[n+1]} &= \left(\mathbf{X}\mathbf{X}^T + \Lambda^*\right)^{-1} \left(\mathbf{Y}\mathbf{X}^T\right)^T; \\ \Lambda^* &= \arg\max_{\Lambda}\{\min_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \Lambda)\} \\ &= \arg\min_{\Lambda} \mathcal{L}_{\mathbf{D}^*}(\Lambda), \end{aligned} \tag{4.27}$$

where $\mathcal{L}_{\mathbf{D}^*}(\Lambda) = \mathrm{trace}\{\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \Lambda)^{-1}\mathbf{X}\mathbf{Y}^T + c\Lambda\}$. The optimization problem based on Λ can not be solved analytically. The objective $\mathcal{L}_{\mathbf{D}^*}(\Lambda)$ is differentiable and a practical method to solve (4.27) is thus to use a gradient descent, or conjugate gradient, method. In this framework the gradient and the Hessian of $\mathcal{L}_{\mathbf{D}^*}(\Lambda)$ can be found using the following formulas,

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathbf{D}^*}(\Lambda)}{\partial \lambda_i} &= c\mathbf{I} - (\mathbf{X}\mathbf{X}^T + \Lambda)^{-1}\mathbf{X}\mathbf{Y}^T\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \Lambda)^{-1} \circ \mathbf{I} \\ \frac{\partial^2 \mathcal{L}_{\mathbf{D}^*}(\Lambda)}{\partial \lambda_i \partial \lambda_j} &= 2(\mathbf{X}\mathbf{X}^T + \Lambda)^{-1}\mathbf{X}\mathbf{Y}^T\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \Lambda)^{-1} \circ (\mathbf{X}\mathbf{X}^T + \Lambda)^{-1} \end{aligned} \tag{4.28}$$

where $\circ$ is the Hadamard (entrywise) product. A difficulty with using (4.28) is the need to calculate the inverse of a large matrix after each update of Λ. Therefore it is difficult to apply this algorithm to the large size problems.

Another dictionary learning method has been presented recently to find a dictionary which minimizes the empirical risk [HH07]. It can use a more general signal generative model, where there is a non-linearity in the model and use a novel technique to solve the optimization problem. In this framework *clean* training samples, i.e. not noisy, are assumed to be available and the empirical risk is defined as the total deviation of the approximated data from training samples. In Hilbert space, an empirical risk can be defined using total estimation error and be found using the following formula [HH07],

$$\mathrm{risk}(\mathscr{Y}, \mathbf{D}) := \frac{1}{2} \sum_{\mathbf{y}_i \in \mathscr{Y}} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2, \tag{4.29}$$

where $\hat{\mathbf{y}}_i$ is the estimation using $(\mathbf{D}, \mathbf{y}_i)$. Here the estimator is the signal approximation oper-

[10] Here a diagonal matrix Λ is admissible, if all the diagonal elements are non-negative.

ator. Horesh *et al.* proposed a generalized generative model as follows,

$$\mathbf{y} = J\mathbf{D}\mathbf{x} + \mathbf{n}, \tag{4.30}$$

where J is an operator which can be non-linear and $\mathbf{n}$ is an *i.i.d* Gaussian vector. Here the standard dictionary learning problem will be investigated and the reader is referred to [HH07] for the general case, when J is not identity operator. When J is identity operator, the minimum risk dictionary is found by solving the following problem,

$$\begin{aligned} \mathbf{D}^* &= \arg\min_{\mathbf{D}} \sum_{\mathbf{y}_i \in \mathscr{Y}} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 \\ s.t. \quad \hat{\mathbf{y}}_i &= \mathbf{D}\hat{\mathbf{x}}_i \\ \hat{\mathbf{x}}_i &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{y}_i\|_2^2 + \lambda\|\mathbf{x}\|_1. \end{aligned} \tag{4.31}$$

The non-differentiability of the forward objective, caused by ℓ_1-norm, does not let us to use standard optimization techniques. Horesh *et al.* used ϵ-relaxed version of the sparsity measure which is defined as follows,

$$\|\mathbf{x}\|_{1,\epsilon} := \sum_{i \in \mathcal{I}} (x_i^2 + \epsilon)^{1/2} \tag{4.32}$$

where $0 < \epsilon \ll \min_{i \in \mathcal{I}} x_i^2$. $\|.\|_{1,\epsilon}$ is a differentiable operator and, when ϵ is selected to being very small, behaves like ℓ_1. The ϵ-relaxed sparsity measures have been used in the sparse approximation [CWB08] and the decoding operator of the non-convex compressed sensing [Cha07, DG09, SY09, FL09]. Now that the forward optimization objective, $\frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{y}_i\|_2^2 + \lambda\|\mathbf{x}\|_{1,\epsilon}$, is differentiable, the optimum $\hat{\mathbf{x}}_i$ can be found by letting the gradient being zero as follows,

$$\mathbf{g}(\hat{\mathbf{x}}_i, \mathbf{D}) = \mathbf{D}^T(\mathbf{D}\hat{\mathbf{x}}_i - \mathbf{y}_i) + \lambda \operatorname{diag}\left(\frac{1}{|\{\hat{x}_i\}_j|_\epsilon}\right)\hat{\mathbf{x}}_i = 0, \tag{4.33}$$

where $|\nu|_\epsilon := \sqrt{\{\nu\}^2 + \epsilon}$. To find $\hat{\mathbf{x}}_i$ from (4.33), it is easier to use an Iterative Reweighted ℓ_2 framework, which is to majorize the sparsity measure in the forward objective with a quadratic term and applying majorization minimization technique, see Subsection 3.4.3.2. At the $n+1$'th Iterative Reweighted ℓ_2 iteration of the Newton's method we solve,

$$\left(\mathbf{D}^T\mathbf{D} + \lambda \operatorname{diag}\left(\frac{1}{|\{\hat{x}_i^{[n]}\}_j|_\epsilon}\right)\right)\delta\hat{\mathbf{x}}_i = -\mathbf{g}(\hat{\mathbf{x}}_i^{[n]}, \mathbf{D}), \tag{4.34}$$

where $\delta\hat{\mathbf{x}}_i$ is the update vector. The update formula $\hat{\mathbf{x}}_i^{[n+1]} = \hat{\mathbf{x}}_i^{[n]} + \delta\hat{\mathbf{x}}_i$ is used for a certain

number of iterations to find the sparse approximations of $\mathbf{y}_i$ for each $i \in \mathcal{I}$.

To update the dictionary, the minimum risk dictionary learning problem can be reformulated as,

$$\begin{aligned} &\arg\min_{\mathbf{D}} \sum_{\mathbf{y}_i \in \mathscr{Y}} \min_{\hat{\mathbf{x}}_i} \|\mathbf{D}\hat{\mathbf{x}}_i - \mathbf{y}_i\|_2^2 \\ &\text{s.t.} \quad \mathbf{g}(\hat{\mathbf{x}}_i, \mathbf{D}) = 0, \ \forall i \in \mathcal{I} \end{aligned} \tag{4.35}$$

(4.35) is a non-linear, equality constrained problem which can approximately be solved by finding a local minimum, using Sequential Quadratic Programming (SQP). Horesh *et al.* used the Newton's method in [HH07] to solve the linearized version of the Lagrangian function $\mathcal{L}$ defined as follows,

$$\mathcal{L}(\{\hat{\mathbf{x}}_i, \lambda_i\}_{i\in\mathcal{I}}, \mathbf{D}) = \sum_{i\in\mathcal{I}} \left(\frac{1}{2}\|\mathbf{D}\hat{\mathbf{x}}_i - \mathbf{y}_i\|_2^2 + \lambda_i \mathbf{g}(\hat{\mathbf{x}}_i, \mathbf{D}) \right) \tag{4.36}$$

where $\{\lambda_i\}_{i\in\mathcal{I}}$ are Lagrangian multipliers. The gradient of $\mathcal{L}$ based on each parameter and the Newton's update are derived in [HH07, Sections 4.1 and 4.2]. As usual, the alternating updates based on $\{\hat{\mathbf{x}}_i\}_{i\in\mathcal{I}}$ and $(\{\hat{\mathbf{x}}_i\}_{i\in\mathcal{I}}, \mathbf{D})$ are calculated for a certain number of iterations.

The reviewed methods here can be modified to learn a structured dictionary. A structured dictionary is generally a dictionary in which the atoms are correlated. Some of the structured dictionary learning methods will be explored in the next section.

4.4 Structured Dictionary Learning Methods

The dictionary learning with a constraint on ℓ_2 or Frobenius norms, was called minimally constrained dictionary learning. In general, applying more constraints on the dictionary can be useful for the following reasons:

1. *Inducing prior information on the dictionary:* In many cases, we do not know about the optimal dictionary but we know that it follows a model or has a property. This can facilitate the dictionary learning by reducing the dictionary search space or reducing computational complexity. Shift-invariance, harmonic type and multiscale are some examples

of such structures.

2. *Finding a dictionary with a fast implementation:* Unstructured dictionaries are rarely used in practice for a simple reason: heavy computation. Inducing appropriate structures on the dictionary might help us to find a dictionary with a fast implementation. The multiscale, sparse and union of orthonormal bases are in this class of structures.

3. *Facilitating the sparse approximations:* Some sparse approximation method can be used with a class of dictionaries, e.g. union of orthonormal basis [SBT00], or might be implemented more efficiently if the dictionary satisfies a property, e.g. being non-singular [MBZJ09].

4. *Reducing the number of training samples:* A set of training samples are given in the dictionary learning problem. If the size of this set is large enough, it indicates the corresponding class of signals and the dictionary learning methods are more successful to find a suitable dictionary in such a setting. Handling a large set of training samples is not easy in general. By applying a structure to the dictionary, we can reduce the number of unknown parameters and thus we need fewer training samples for the dictionary learning. The sparse or compressible dictionary learning [?, YD09] and parametric dictionary learning are some examples of such structured dictionary learning methods.

Some of these structures and corresponding dictionary learning methods have been briefly explored in the following. These methods are basically based on modifying standard dictionary learning methods, explored in Section 4.3, to preserve the structure of dictionary, during learning process.

4.4.1 Shift Invariant Dictionary Learning

A large class of natural signals is presented by time series, e.g. audio, video and Electrocardiography (ECG). An analysis of time series should typically be time-shift independent. For example the start time of a sound track does not change the identity of the track. To have a shift-resilient sparse approximation, the dictionary has to include the shifted versions of a set of atoms, called mother dictionary here, at each time instance. The dictionary size is thus very large and the atoms are very correlated in general, i.e. the inner products between atoms, which present the similarities between atoms, are large. The dictionary learning and sparse approximation, using such a dictionary, is challenging.

Let the ith mother atom, the mother dictionary and the time-shift operator respectively be $\mathbf{d}_i$, $\mathbf{D}_g = [\mathbf{d}_i]_{i\in\mathcal{I}_g}$ and Γ_{t_s}, where $\mathcal{I}_g$ is the set of indices and t_s is the time-shift. Note that Γ_{t_s} can be a circular shift operator, in which the off-windowed points of the mother atoms appear at the beginning of the new atom. A set of time shifts $\mathcal{T}$ should be selected to generate the dictionary $\mathbf{D}$. The largest $\mathcal{T}$ for a time series[11] is $\mathcal{T} = \{t_s : 0 \leq t_s \leq d-1\}$, where d is the block size[12]. The dictionary now can be generated using time shift operator as follows,

$$\mathbf{D} = [\Gamma_{t_s}\mathbf{d}_i]_{i\in\mathcal{I}_g, t_s\in\mathcal{T}}. \tag{4.37}$$

The size of the dictionary is approximately $|\mathcal{I}_g|.|\mathcal{T}|$, which is often significantly larger than the mother dictionary $\mathbf{D}_g$. The structure of such a dictionary allowes us to represent the matrix-vector multiplications by convolution operator. The convolution operator can be implemented efficiently using filter-banks.

The shift-invariant dictionary learning problem is simplified by assuming $\mathcal{T}$ is given. The problem is to find the mother dictionary in this setting. Let the conjugate operator of Γ_{t_s} be defined as the operator in which $\forall\mathbf{d}, \forall\mathbf{y},\ \langle\Gamma_{t_s}\mathbf{d}, \mathbf{y}\rangle = \left\langle\mathbf{d}, \Gamma^*_{t_s}\mathbf{y}\right\rangle$. This operator is easily found by Γ_{-t_s} which is time shift with t_s, in the reverse direction. There are two approaches to handle such a dictionary learning problem, as follows,

1. *Using* $\mathbf{D} = [\Gamma_{t_s}\mathbf{d}_i]_{i\in\mathcal{I}_g,\ t_s\in\mathcal{T}}$ *and* $\mathscr{Y} = \{\mathbf{y}_l\}_{1\leq l\leq L}$*:* In this approach the size of the dictionary is very large and a fast sparse approximation technique should be used. It is also possible to restrict the search space in the sparse approximation by only considering the atoms which are more correlated to $\mathscr{Y}$, at the first iteration, for the following iterations of the greedy sparse approximation methods [BD06].

2. *Using* $\mathbf{D} = \mathbf{D}_g = [\mathbf{d}_i]_{i\in\mathcal{I}_g}$ *and* $\mathscr{Y} = \{\Gamma^*_{t_s}\mathbf{y}_l\}_{1\leq l\leq L,\ t_s\in\mathcal{T}}$*:* The multiplication of a signal with a shifted-atom can be equivalently calculated by multiplying the reverse-shifted signal and the atom. In this setting the dictionary size is small, N, and the number of samples is increased by a factor of $|\mathcal{T}|$. Although the conventional sparse approximation methods can be used in the first step, the dictionary update is more computationally intensive.

[11] If the proposed class of signals is made by sampling analog signals, the mother atom can be analog and the set of time shifts can be larger.

[12] Sometimes the upper limit is chosen being $d + \Delta - 1$, where Δ is the width of the mother atom.

The first approach has often been selected by the researchers [LS98, BD06, JLVG06, MLGB08, GRKN07]. The second approach has been recently investigated in [VT09]. Note that although these two settings are proposed for the same problem, the solutions might be different in general.

The algorithm finds or updates the sparse approximations $\mathbf{x}_l$ in the first step. As mentioned earlier, when the dictionary size is large, specially in the first approach, a modified sparse approximation method, see for example [KG06] and [BD06], or a convolutive sparse approximation method [FBSJ08] can be used to handle such a dictionary. The difference between these methods are mainly in the second step of the alternating minimization, which is dictionary update.

One method is to use a technique similar to the dictionary learning based on ML, see subsection 4.3.1. This has been used in [BD06], in which the dictionary update have been found by applying the stochastic gradient method to solve the ML problem. In this framework the dictionary is updated in the negative gradient of approximation error.

Another method is to extend the K-SVD dictionary learning method, see subsection 4.3.4, to the shift-invariance framework [MLGB08, VT09]. The extension of K-SVD for the second approach is easily derived as long as its formulation is similar to the unstructured dictionary learning problem. The size of the problem in the second approach is significantly large and an efficient implementation of K-SVD [RZE08] has to be used. The extension to the first approach is more difficult in general, which can be simplified if the shifted atoms are not overlapped [Les07]. In this framework, all but one of the mother atoms are fixed and the approximation error is minimized by updating the selected mother atom and the corresponding *rows* of coefficient matrix. Let the ith mother atom $\mathbf{d}_i$ have to be updated. The corresponding objective can be presented as follows[13],

$$(\mathbf{d}_i^{[n+1]}, \{\mathbf{x}_{t_s}^{(i)^{[n+1]}}\}_{t_s \in \mathcal{T}}) = \arg \min_{(\mathbf{d} \in \mathcal{D}, \bar{\mathbf{x}}_{t_s} \in \mathbb{R}^N, \forall t_s \in \mathcal{T})} \|\mathbf{E}_{\mathcal{T} \setminus i} - \sum_{t_s \in \mathcal{T}} \Gamma_{t_s}\{\mathbf{d}\}\, \bar{\mathbf{x}}_{t_s}^T\|_F^2 + \lambda \sum_{t_s \in \mathcal{T}} \|\bar{\mathbf{x}}_{t_s}\|_0, \tag{4.38}$$

[13] Note that [Les07] and [MLGB08] present a slightly different formulation for this problem. Lesage and Mailhe *et al.* assumed a long training signal for the dictionary learning. Although this assumption is valid in special cases, here a more general case, which is more closed to the standard K-SVD formulation, is derived. This formulation is more helpful when a set of training samples are given or the signal is multichannel, for example the stereo audio and multichannel ECG.

where $\mathbf{E}_{\mathcal{I}\setminus i}$ here represents the approximation error caused using all atoms but $\{\Gamma_{t_s}\mathbf{d}_i\}_{t_s\in\mathcal{T}}$, $\bar{\mathbf{x}}_{t_s}$ represent the coefficients corresponding to $\Gamma_{t_s}\mathbf{d}$ and $\{\mathbf{x}_{t_s}^{(i)}\}_{t_s\in\mathcal{T}}$ includes all rows of the coefficient matrix $\mathbf{X}$, which are related to $\mathbf{d}_i$. If $\Gamma_{t_s}\mathbf{d}$'s are not overlapped, (4.38) can be reformulated using the conjugate operator $\Gamma_{t_s}^*$ as follows,

$$(\mathbf{d}_i^{[n+1]}, \{\mathbf{x}_{t_s}^{(i)^{[n+1]}}\}_{t_s\in\mathcal{T}}) = \arg \min_{(\mathbf{d}\in\mathcal{D}, \bar{\mathbf{x}}_{t_s}\in\mathbb{R}^N, \forall t_s\in\mathcal{T})} \sum_{t_s\in\mathcal{T}} \|\Gamma_{t_s}^*\mathbf{E}_{\mathcal{I}\setminus i} - \mathbf{d}\,\bar{\mathbf{x}}_{t_s}^T\|_F^2 + \lambda\|\bar{\mathbf{x}}_{t_s}\|_0, \tag{4.39}$$

where $\Gamma_{t_s}^*$ is now applied on each column of $\mathbf{E}_{\mathcal{I}\setminus i}$. This reformulation was derived using the fact that the matrices $\Gamma_{t_s}\{\mathbf{d}\}\,\bar{\mathbf{x}}_{t_s}^T$ for different t_s are orthogonal, in Hilbert space defined on the matrix space using $\langle\mathbf{A},\mathbf{B}\rangle = \mathrm{tr}\{\mathbf{A}^T,\mathbf{B}\}$ as the inner-product, for non-overlapped shifts. The optimization problem (4.39) is difficult to solve based on both parameters, $\mathbf{d}$ and $\bar{\mathbf{x}}_{t_s}$. A solution can be to use the block relaxation technique and solve based on each parameter, when the other parameter is kept fixed, as it has been proposed for K-SVD dictionary learning in [RZE08]. The update of $\mathbf{d}$ can be found using the Projection Onto the Convex Sets (POCS) method [GPR67, CT90]. The update of each $\bar{\mathbf{x}}_{t_s}$ can be found by minimizing ℓ_2-norm approximation error in (4.39), while keeping the support of $\bar{\mathbf{x}}_{t_s}$ fixed, see [RZE08] for more detail.

If the shifted atoms have overlaps, the above method is not accurate. Mailhe *et al.* [MLGB08] proposed a modification to the dictionary update step by introducing an extra weighting matrix to penalize the overlapped part of the signals based on the number of overlapped atoms at that point.

The K-SVD method was shown to be computationally expensive for the large scale problems. Although the size of dictionary is very large here, the implementation of the shift-invariant K-SVD is just slightly more expensive than K-SVD as the actual size of problem in (4.39) is moderate.

Another approach for the shift-invariant dictionary learning is to use the convolution operator, which allowes us to update the dictionary in Fourier domain. If $\mathcal{T}$ includes all possible discrete time shifts and the shift operator is circular, the dictionary $\mathbf{D}$ can be written as,

$$\mathbf{D} = [C(\mathbf{d}_i)]_{i\in\mathcal{I}} \in \mathbb{R}^{d\times dN}, \tag{4.40}$$

where $C(\mathbf{d}_i) \in \mathbb{R}^{d\times d}$ is the circulant matrix [Gra06] parametrized by the vector $\mathbf{d}_i$, which is

defined as follows,

$$C(\mathbf{d}_i) = \begin{bmatrix} d_{1,i} & d_{d,i} & \dots & d_{3,i} & d_{2,i} \\ d_{2,i} & d_{1,i} & \dots & d_{4,i} & d_{3,i} \\ \vdots & & \ddots & & \vdots \\ d_{d-1,i} & d_{d-2,i} & \dots & d_{1,i} & d_{d,i} \\ d_{d,i} & d_{d-1,i} & \dots & d_{2,i} & d_{1,i} \end{bmatrix}, \tag{4.41}$$

where $d_{j,i}$ is the jth component of $\mathbf{d}_i$. The generative model can be reformulated using the circular convolution operator, "$*$", as follows [LS98],

$$\mathbf{y} \approx \sum_{i \in \mathcal{I}} \mathbf{x}_i * \mathbf{d}_i. \tag{4.42}$$

The shift-invariant dictionary learning is thus reformulated as follows,

$$\{\mathbf{d}_i^*, \mathbf{x}_{i,l}^*\}_{i \in \mathcal{I}, 1 \le l \le L} = \arg \min_{\substack{\{\mathbf{d}_i\}_{i \in \mathcal{I}}, \\ \{\mathbf{x}_{i,l}\}_{i \in \mathcal{I}, 1 \le l \le L}}} \sum_{1 \le l \le L} \left(\| \sum_{i \in \mathcal{I}} \mathbf{x}_{i,l} * \mathbf{d}_i - \mathbf{y}_l \|_2^2 + \lambda \sum_{i \in \mathcal{I}} \|\mathbf{x}_{i,l}\|_1 \right). \tag{4.43}$$

The optimal dictionary can be found by $\mathbf{D}_g^* = [\mathbf{d}_i^*]_{i \in \mathcal{I}}$. The alternating minimization technique has been chosen to solve (4.43) in [LS98] and [GRKN07]. The coefficients can be updated using the method explained earlier. Lewicki *et al.* [LS98] optimized the objective in (4.43), with respect to $\{\mathbf{d}_i\}_i \in \mathcal{I}$ while the coefficients are kept fixed, using a gradient descent method. Grosse *et al.* [GRKN07] chose a different path to find the dictionary update. Each atom appears at many instances in the formulation (4.43), which makes it difficult to optimize based on $\mathbf{D}_g$. The gradient descent method for such a problem does not converge fast. The authors of [GRKN07] used the Parseval's theorem and solved the problem in the Fourier domain. The quadratic terms of (4.43), which depend on the dictionary, in the Fourier domain are as follows,

$$\sum_{1 \le l \le L} \| \sum_{i \in \mathcal{I}} \left\langle \hat{\mathbf{x}}_{i,l}, \hat{\mathbf{d}}_i \right\rangle - \hat{\mathbf{y}}_l \|_2^2 \tag{4.44}$$

where "$\hat{}$" shows the value of the parameter in the Fourier domain. Minimization of (4.44) with respect to the dictionary is significantly easier than (4.43) and it has been done using Lagrangian multiplier method while the atoms are constrained being in the convex admissible set $\|\mathbf{d}_i\|_2^2 \le 1$, $\forall i \in \mathcal{I}$. They then used Newton's method to solve the dual problem, similar to the method was explained in subsection 4.3.5.

The shift-invariant dictionary learning problem can also be solved using greedy methods. In a greedy sparse approximation often the set of selected atoms is increased by adding some atoms which are more corrected to the residual of the signal, for example see the MP method in subsection 3.3.1. In this framework, each atom can be updated to have more correlation with the given training samples in the dictionary update stage. An extra constraint is induced in [JLVG06] to find an incoherent dictionary and to prevent the existence of two similar atoms in the dictionary. In this framework, which is called Matching of Time Invariant Filters (MoTIF) in [JLVG06], the first updated atom is found by solving the following problem,

$$\mathbf{d}_1^{[n+1]} = \arg \max_{\|\mathbf{d}\|_2=1} \sum_{1\leq l\leq L} \max_{t_p\in\mathcal{T}} |\left\langle \mathbf{y}_l, \Gamma_{t_p}\mathbf{d}\right\rangle|^2, \tag{4.45}$$

and by penalizing the inner-product between the atom and the currently selected atoms, the new atoms can be found by minimizing the following problem,

$$\mathbf{d}_i^{[n+1]} \overset{2\leq i\leq N}{=} \arg \max_{\|\mathbf{d}\|_2=1} \frac{\sum_{1\leq l\leq L} \max_{t_p\in\mathcal{T}} |\left\langle \mathbf{y}_l, \Gamma_{t_p}\mathbf{d}\right\rangle|^2}{\sum_{1\leq j<i} \sum_{t_p\in\mathcal{T}} |\left\langle \mathbf{d}_j, \Gamma_{t_p}\mathbf{d}\right\rangle|^2}. \tag{4.46}$$

Solving (4.45) and (4.46) are not easy and Jost *et al.* in [JLVG06] proposed a two step optimization technique to simplify the problem. In the first step, the best t_p is found while $\mathbf{d}$ is kept fixed and the next step $\mathbf{d}$ is updated, while t_p is kept fixed. The optimum time shift t_p^* is found, in the first step, by checking all $t_p \in \mathcal{T}$. The second step is more complicated, which can be simplified using the conjugate operator Γ^*. Using the fact that $\left\langle \mathbf{y}_l, \Gamma_{t_p}\mathbf{d}\right\rangle = \left\langle \Gamma_{t_p}^*\mathbf{y}_l, \mathbf{d}\right\rangle$, the update of $\mathbf{d}_1$ can be found by solving the following problem,

$$\mathbf{d}_1^{[n+1]} = \arg \max_{\|\mathbf{d}\|_2=1} \mathbf{d}^T\mathbf{A}\mathbf{d}, \tag{4.47}$$

where $\mathbf{A} = \mathbf{F}\mathbf{F}^T$, in which $\mathbf{F} = [\Gamma_{t_p^*}^*\mathbf{y}_l]_{1\leq l\leq L}$. The optimal atom in (4.47) is the normalized eigenvector associated with the significant eigenvalue of $\mathbf{A}$.

The conjugate operator Γ^* can also be used to simplify (4.46). Let $\mathbf{B}$ be defined as, $\mathbf{B} = \sum_{1\leq j<i} \sum_{t_p\in\mathcal{T}} \Gamma_{t_p^*}^*\mathbf{d}_j \, (\Gamma_{t_p^*}^*\mathbf{d}_j)^T$. (4.46) is now reformulated as follows,

$$\mathbf{d}_i^{[n+1]} \overset{2\leq i\leq N}{=} \arg \max_{\|\mathbf{d}\|_2=1} \frac{\mathbf{d}^T\mathbf{A}\mathbf{d}}{\mathbf{d}^T\mathbf{B}\mathbf{d}}. \tag{4.48}$$

The best atom $\mathbf{d}_i^{[n+1]}$ is the normalized eigenvector associated to the significant eigenvalue

of the *generalized* eigenvalue problem [GVL96] (4.48). Finally it is worth mentioning that because there exists no explicit objective to optimize in the dictionary update step, the convergence analysis of the algorithm is challenging.

In the next subsection another useful structure on the dictionary, called a multiscale structure, will be introduced. This structure helps to implement the dictionary-coefficient multiplications more efficiently. It also induces prior information about the generating model to the dictionary.

4.4.2 Multiscale Dictionary Learning

Multiscale transforms were found to be successful in representing some class of signals sparsly, e.g. images and videos, see for example [Mal99]. Wavelet and multiscale Gabor transforms are some examples of this type of transforms. Such prior information about the model can be induced in the dictionary. This constraint, in the simplest case, can be induced on dictionary using a dyadic structure. In this framework a mother wavelet generates all the atoms using the down sampling by a factor of two and the time shifting operators. This is a more tight constraint on the dictionary than only shift invariance, in which the atoms are not supposed to be a scaled copy of one, or more, generative mother wavelet(s).

The dictionary learning problem has been solved subject to the multiresolution constraint in a stochastic framework in [SO03]. Sallee *et al.* [SO03] chose the MAP estimation framework to find the sparse coefficients and the maximum log-likelihood method to update the dictionary. A prior distribution for the coefficients is chosen to be a mixture of a delta function at zero and a Gaussian elsewere, which promotes sparsity of approximations, and they also used a Gaussian additive noise generative model with a multiscale dictionary model. To find the sparse coefficients, they used a Gibbs sampler to sample the posterior distribution. A gradient ascent method has been used in the dictionary update stage, which is similar to the method explored previously, in subsection 4.3.1, for the dictionary learning based on ML.

The multiscale structure can be relaxed using atoms with different support sizes and not directly derived using mother atoms. In other words, it follows a tree structure, which can particularly be dualtree and quadtree. Although the new model preserves some features of the multiscale dictionaries, it can be learned easier by applying a shift invariant dictionary learning to each individual scale. [MSE08] chose such a structure and learned the dictionary using the K-SVD method.

4.4.3 Unions of Orthonormal Bases Dictionary Learning

One method to generate an overcomplete dictionary is to concatenate some orthonormal bases. This structure has been used to accelerate some sparse approximation methods, see for example [SBT00, BST98]. This structure have also been used in morphological component analysis [SED04, SED05, ESQD05]. Therefore it is relevant to investigate how this structure can be applied to the dictionary learning problem. This structure is a generalization of the principal component analysis (PCA) [Jol02], called GPCA [VMS05], in which the aim is to find some orthonormal bases which represent the principal components. PCA relates to the singular vectors of the training matrix $\mathbf{Y}$. It is extended to a dictionary learning framework in [LGBB05], where the coefficients need to be sparse. This framework will be explored here.

Let the aim primarily be to find the best orthogonal dictionary $\mathbf{D} \in \mathbb{R}^{d\times d}$, for a given set of training samples $\mathscr{Y}$. The first stage as usual is to find the sparse approximation $\mathbf{X}^{[n+1]}$ using the current dictionary $\mathbf{D}^{[n]}$[14]. In the dictionary update stage, the following optimization problem should be solved,

$$\mathbf{D}^{[n+1]} = \arg\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{[n+1]}\|_F^2, \tag{4.49}$$

where $\mathcal{D}$ is the set of orthonormal matrices in $\mathbb{R}^{d\times d}$. Let $\Lambda = [\lambda_{i,j}]_{i,j\in[1,d]} : \lambda_{i,j} \in \mathbb{R}_0^+$ be the Lagrangian multiplier. By using the Lagrange multipliers method, (4.49) can be solved by setting the gradient of the following Lagrangian function to zero,

$$\mathcal{L}(\mathbf{D}, \Lambda) = \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{[n+1]}\|_F^2 + \mathrm{tr}\{\Lambda(\mathbf{D}^T\mathbf{D} - \mathbf{I})\}. \tag{4.50}$$

It has been shown, for example in [LGBB05, Appendix A], that the solution of the above problem can be found by $\mathbf{D}^{[n+1]} = \mathbf{U}\mathbf{V}$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is one of the singular value decompositions of $\mathbf{Y}{\mathbf{X}^{[n+1]}}^T$.

This can be generalized to a union of the K orthonormal bases by letting $\mathbf{D}$ be $[\mathbf{D}_i]_{i\in[1,K]}$, i.e. $\mathbf{D}$ is made by concatenation of $\mathbf{D}_i$'s. The dictionary update step, subject to this structure, can be done by minimizing the following optimization problem,

$$\mathbf{D}^{[n+1]} = [\mathbf{D}_i^*]_{i\in[1,K]} = \arg\min_{\mathbf{D}_i|_{i\in[1,K]}\in\mathcal{D}} \|\mathbf{Y} - [\mathbf{D}_i]_{i\in[1,K]}\mathbf{X}^{[n+1]}\|_F^2, \tag{4.51}$$

There exists no easy method to solve (4.51) exactly. A tractable method is the block-relaxed

[14] Note that in this setting, $\mathbf{X}^{[n+1]}$ is easily found by ${\mathbf{D}^{[n]}}^T\mathbf{Y}$ which might not be sparse.

minimization method, in which all but one $\mathbf{D}_i$ is fixed at a time. The update of $\mathbf{D}_i$ is found using Lagrange multipliers method as introduced in the orthogonal dictionary learning case. Therefore the dictionary update stage includes K steps of the $\mathbf{D}_i$ updates.

4.4.4 Other Structured Dictionary Learning Methods

Structures introduced in the previous subsections include most of the structures that have been used for the dictionary learning. There are other structures for the dictionary learning which have not been explored as much. These structures are discussed next.

4.4.4.1 Block-Overlapped Dictionary Learning (BODL)

An issue with the sparse approximation of a time series in a block-based framework, i.e. using a generative model like (2.1), is the edging effects. It means that the time series has some artefacts, which often appears as a jump in the time series, at the connections of consecutive blocks. This fact has also been observed in the orthogonal transform approximations of the signals, for example audio. One solution in the orthogonal approximation is to use lapped transforms, like MDCT and MCLT [Mal92]. These transforms are the generalization of the standard orthogonal transforms by letting the windows of the transforms have some overlap. The overcomplete dictionaries can also be modified in this framework to overlap and be used for sparse approximations [KG06]. A question is now how to learn such a dictionary? Engan *et al.* propose a framework in [ESH07] for the block-overlapped dictionary learning. In this framework the dictionary $\mathbf{D}$ is a *block-diagonal* matrix with an overlapping dictionary $\Phi \in \mathbb{R}^{d\times N}$ as the diagonal blocks, i.e. $\mathbf{D} = \mathrm{diag}\{\overbrace{\Phi, \Phi, \ldots, \Phi}^{K}\}$. Note that $\mathbf{D}$ represents a block overlapped dictionary when $K \to \infty$. Therefore with a finite K we only *approximate* such a dictionary and the optimum dictionary is sub-optimal. Let any Φ_i and Φ_{i+1} have $d - P$ overlapping points which is assumed $d = RP$ for simplicity. The generative model can be formulated as follows,

$$\mathbf{y} = \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{P,1} & \cdots & \phi_{d,1} & & & & & \\ \vdots & & \vdots & & \vdots & & & & 0 & \\ \phi_{1,N} & \cdots & \phi_{P,N} & \cdots & \phi_{N,d} & & & & & \\ & & \phi_{1,1} & \cdots & \phi_{P,1} & \cdots & \phi_{d,1} & & & \\ & & \vdots & & \vdots & & \vdots & & & \\ & & \phi_{1,N} & \cdots & \phi_{P,N} & \cdots & \phi_{N,d} & & & \\ & & & & \phi_{1,1} & \cdots & \phi_{d-P,1} & \cdots & \phi_{d,1} & \\ & & & & \vdots & & \vdots & & \vdots & \\ & & & & \phi_{1,N} & \cdots & \phi_{d-P,N} & \cdots & \phi_{N,d} & \\ & 0 & & & & & & & & \ddots \end{bmatrix}^T \mathbf{x}. \tag{4.52}$$

Φ can be updated to minimize the approximation error in the dictionary update step while $\mathbf{X}$ is kept fixed. The dictionary update can be found using a method similar to MOD, see subsection 4.3.3, where the parameters are slightly modified. If we partition Φ into R subdictionaries $\Phi_r \in \mathbb{R}^{P\times N}$, then $\Phi = [\Phi_r^T]_{r\in[1,R]}^T$. The training sample $\mathbf{y}$, which is assumed to be a time series, is partitioned with the block size P, $[\hat{\mathbf{y}}_l]_{l\in[1,L]}$ and the generative model is reformulated as,

$$\overbrace{[\hat{\mathbf{y}}_1 \dots \hat{\mathbf{y}}_l \dots \hat{\mathbf{y}}_L]}^{\hat{\mathbf{Y}}} = \overbrace{[\Phi_1 \dots \Phi_r \dots \Phi_R]}^{\hat{\Phi}} \overbrace{\begin{bmatrix} x_1 & \cdots & x_l & \cdots & x_L \\ x_0 & \cdots & x_{l-1} & \cdots & x_{L-1} \\ \vdots & & \vdots & & \vdots \\ x_{1-RN} & \cdots & x_{l-RN} & \cdots & x_{L-RN} \end{bmatrix}}^{\hat{\mathbf{X}}} \tag{4.53}$$

where x_i is the ith element of the coefficient time series $\mathbf{x}$, which is found by some sparse approximation method using $\mathbf{D}$ as the dictionary. The dictionary update is now found by solving the following optimization problem,

$$\hat{\Phi}^* = \arg \min_{\mathbf{D}\in\mathbb{R}^{\mathbf{R}\times\mathbf{RN}}} \|\hat{\mathbf{Y}} - \hat{\Phi}\hat{\mathbf{X}}\|_F^2, \tag{4.54}$$

which is similar to the problem solved in MOD (4.20). The solution of (4.54) can be found

using pseudoinverse operator as follows,

$$\hat{\Phi}^* = \hat{\mathbf{Y}}\hat{\mathbf{X}}^{\dagger}. \tag{4.55}$$

Finding $\mathbf{\Phi}$ by using $\hat{\mathbf{\Phi}}^*$ is trivial. The algorithm can be summarized as iteratively finding the sparse approximation of the training time series $\mathbf{Y}$, using the current Φ, followed by update Φ using the method explained here, while keeping $\mathbf{X}$ fixed.

***Remark* 4.4.1.** Although BODL is presented to facilitate the sparse approximation of time series, or reducing the edging effect, it is a particular case of the shift-invariant dictionary learning which was introduced in subsection 4.4.1, where the time-shift interval is chosen to be more than 1, or more precisely P. Most of the algorithms introduced in that subsection can also be used to learn the dictionary in this framework.

***Remark* 4.4.2.** Another way to reduce the edging effect is to use the framework introduced in [EA06] and [PE09]. Elad *et al.* [EA06] introduced an extra parameter $\hat{\mathbf{y}}$, which approximates the signal $\mathbf{y}$, and then found the dictionary by solving the following optimization problem,

$$\min_{\mathbf{D},\mathbf{x},\hat{\mathbf{y}}} \left\{ \mu\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2 + \lambda\mathcal{J}(\mathbf{x}) + \sum_{l\in[1,L]} \|\mathcal{R}_l\{\hat{\mathbf{y}}\} - \mathbf{D}\mathcal{R}_l\{\mathbf{x}\}\|_2^2 \right\} \tag{4.56}$$

where the operator $\mathcal{R}_l\{.\}$ chooses the lth block of the time series $\mathbf{x}$ and $\hat{\mathbf{y}}$. (4.56) is optimized in an alternating minimization framework in [EA06]. In one step the objective is minimized based on $(\mathbf{D},\mathbf{x})$ using K-SVD algorithm and in the other step $\hat{\mathbf{y}}$ is updated using regularized pseudoinverse operator. See [EA06] for more details about the algorithm and some simulation results.

4.4.4.2 Dictionary Learning using a Signature-Dictionary

Another model for generating a dictionary is to assume that each atom is generated by selecting a part of a signature dictionary, which is simply a patch of signal [AE08]. The aim of this dictionary learning is to learn the signature dictionary. The signature dictionary model reduces the size of the problem, as its size is significantly smaller than the generated dictionary. It helps us to reduce the number of necessary training samples. Although the framework presented in [AE08] is for the sparse approximation of the images, it can be used for one dimensional signals, as it is derived here. Let $\mathbf{g}$ be the signature dictionary and the operator $\mathcal{R}_i\{.\}$ selects

a segment of the operand by windowing in which i represents the window center. $\mathcal{R}_i\{.\}$ is circular, which means it uses a circular window in the edges of the input signal. The dictionary $\mathbf{D}$ is generated using a set of window centers $\mathcal{I}$ as follows, $\mathbf{D} = [\mathcal{R}_i\{\mathbf{g}\}]_{i\in\mathcal{I}}$. The dictionary learning problem can be formulated as before using structured dictionary $\mathbf{D}$ in (4.2). In the dictionary update step of the alternating minimization method, $\mathbf{D}$ is refined using a gradient descent method in [AE08].

4.4.4.3 Sparse Dictionary Learning

A disadvantage of the unstructured dictionary learning methods is that the learned dictionaries can not be implemented efficiently. In the previous subsections it has been shown that how a structure in the dictionary can facilitate the matrix-vector multiplications using one or more filter banks, see subsection 4.4.1. Here another useful structure is introduced which breaks the dictionary-coefficient multiplications into two steps which can be implemented very efficient. The dictionary in this framework is generated using a mother dictionary $\mathbf{\Phi}$ and a sparse matrix $\mathbf{\Psi}$ as follows,

$$\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi} \tag{4.57}$$

The dictionary learning is now to find sparse matrix $\mathbf{\Psi}$. If the dictionary is sparse, according to the generative model (4.57), any dictionary-coefficient vector multiplication can be done by multiplying the coefficient vector and $\mathbf{\Psi}$ in the first step followed by multiplying with $\mathbf{\Phi}$. $\mathbf{\Psi}$ is a sparse matrix and multiplication with such a matrix can be done very efficiently, as long as we only need to multiply the non-zero components of $\mathbf{\Psi}$ with some values of the coefficient vector. On the other hand, by choosing a structured $\mathbf{\Phi}$, with a complexity of at most $\mathcal{O}(N \log N)$, the overall complexity of multiplying with $\mathbf{D}$ reduces to the complexity of $\mathcal{O}(N \log N)$, if $\mathbf{\Psi}$ is very sparse. The computation complexity reduction is explored in more detail in [RZE08].

Rubinstein *et al.* proposed an alternating minimization framework to find the coefficient matrix $\mathbf{X}$ and the sparse generator matrix $\mathbf{\Psi}$. A greedy sparse approximation method is used in each minimization step. Although some promising results are reported in [RZE09], more investigations on the convergence issues and the optimization methods is necessary.

This framework will be explored in more detail in Chapter 7, where the compressibility of $\mathbf{\Psi}$ is proposed. The problem subject to this constraint is more relaxed, which lets the learned dictionary be sparser. It is also shown that the dictionary learning problem is a well-defined

optimization problem. It is followed by introducing a novel optimization method which is guaranteed to converge to a local minimum.

4.5 Summary

This chapter introduced a brief overview on the dictionary learning problem. The problem was introduced in a general framework which has explicitly, or implicitly, been used in most of the dictionary learning methods. Various dictionary learning methods for sparse approximations were then reviewed, while differences and similarities were being emphasized. Sometimes it is necessary to induce a structure on the dictionary. The dictionary learning subject to such a structure has been explored here and it was shown that it can be learned by solving a constrained optimization problem. Some practical optimization methods had been presented after introducing each optimization problem. The relation between structures was also explored and it was shown how the standard dictionary learning methods can be modified to solve such constrained dictionary learning problems. Two structures on the dictionaries, compressible and minimum size, will be introduced in the following chapters and some practical algorithms will also be introduced for each method.

Chapter 5

Dictionary Learning with the Majorization Minimization Method

5.1 Introduction

The dictionary learning is often a large scale multivariable optimization problem. In Chapter 4, it was shown how it can be solved approximately. The dictionary learning methods use the block relaxation technique to simplify the multivariable problem. Here, a new dictionary learning method based on the majorization minimization method is introduced. The new method is scalable and flexible enough to handle different constraints on the dictionaries. The scalability is provided by its simple iterative updates, which can efficiently be implemented using multi-core processors. The flexibility of the algorithm can be used to apply different constraints on, for example, size and norm of the dictionaries. These constraints on the dictionaries will be explored in Chapter 6 and 7.

The given dictionary learning method can use different admissible sets on the dictionaries. The analysis of the algorithms is easier if we use a convex admissible set. Two different admissible sets have been used here, which are convex hulls of the fixed Frobenius and fixed column norm admissible sets.

Another advantage of the proposed dictionary learning method is that it is guaranteed to converge[1]. Such an analysis is difficult or impossible for the algorithms presented in Chapter 4. Implementation of a dictionary learning method would also be easier if the convergence of the algorithm is proved, where there would be no need for an extra procedure to monitor the convergence status of the algorithm.

Using simulation it has been shown that the new dictionary learning method can handle large problems, which are very difficult to be solved by, for example, K-SVD. Latter, it will also be

[1]The convergence here has a slightly different definition to its standard definition in mathematical analysis [Rud76]. The convergence analysis is explored in Appendix B

shown that the learned dictionary is more suitable for the sparse audio coding, when an entropy encoder is used.

5.2 Dictionary Learning using Majorization Minimization

The majorization minimization technique was shown in Subsection 3.4.1 to be a useful approach to solve sparse approximation problem. It was also shown that often dictionary learning is broken down into two optimization problems in a block-relaxation framework, see Chapter 4. The optimization stage of the dictionary update step has similarities with the sparse approximation problem, which is a motivation for applying such a technique to the dictionary learning problem. Note that the optimization problem is now constrained. Let the dictionary learning problem be defined as the following constrained optimization problem,

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \phi(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D} \\ \phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda \mathcal{J}_p(\mathbf{X}), \end{aligned} \tag{5.1}$$

where $\mathscr{D}$ is an admissible set of dictionaries and $\mathcal{J}_p(.)$ is the sparsity measure defined in (2.18). (5.1) is a particular case of the formulation presented in (4.2), by choosing $\mathcal{J}_p(.)$ as the sparsity measure. As noted in [KMR+03], two typical constraints are the unit Frobenius-norm and the unit column-norm constraints, both of which lead to non-convex solution sets. Instead of using these constraints, we can use the convex relaxed version of these constrained sets. These are the convex sets of matrices with bounded Frobenius norm,

$$\mathscr{D}_F = \{\mathbf{D}_{d\times N} : ||\mathbf{D}||_F \leq c_F^{1/2}\} \tag{5.2}$$

where c_F is a constant and the convex set of matrices with bounded column norm,

$$\mathscr{D}_C = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 \leq c_C^{1/2}\}, \tag{5.3}$$

where $\mathbf{d}_i$ is the i$^{\text{th}}$ column of the dictionary $\mathbf{D}$ and c_C is a constant. (5.3) has also been used in [LBRN07] for dictionary learning using the Newton's method, see Subsection 4.3.5. When the sparsity measure in the sparse approximation step penalizes the coefficients based on their magnitudes (e.g. $l_p : 0 < p \leq 1$), it is easy to show that the solution of (5.1) is on the boundary of these convex admissible sets. However, the convex admissible sets also allow the

optimization algorithm to "pass through" these admissible sets while the traditional non-convex sets only allow the algorithm to move along the boundary of these sets.

Like other dictionary learning methods, a block relaxation technique, see for example [Lee94], is used to solve (5.1), where $p = 1$. There are two blocks of parameters, $\mathbf{X}$ and $\mathbf{D}$, in this framework which can be minimized alternatingly in this framework. The alternating minimization continues until the algorithm converges to an accumulation point. For a fixed dictionary, ℓ_1 penalized sparse approximation is a convex optimization problem and using convex dictionary admissible sets also turns the dictionary update into a convex optimization problem. Whilst this allows us to find the optimum update in each step, (5.1) is not convex as a function of the pair $(\mathbf{X}, \mathbf{D})$, and alternating optimization is not guaranteed to find a global optimum.

Various methods have been presented to solve the ℓ_1 penalized sparse approximation [CDS98, EHJT04, DDD04]. The Iterative Thresholding approach, see Chapter 3, has been chosen here. This method is extended to the sparse matrix approximation problem (2.19) in section 5.2.1.

In the next subsections it is shown that the majorization minimization method can be used to optimize the objective introduced in (5.1) based on $\mathbf{X}$ (Subsection 5.2.1) or $\mathbf{D}$ (Subsections 5.2.2) using different constraints. Updating the coefficient or the dictionary matrices always reduces the joint objective function or keeps it at the same value. The fact that the objective function is lower-bounded is sufficient to show stability of the updating process in the sense of Lyapunov (Lyapunov second theorem) [Lya66]. A basic convergence proof for the proposed algorithm is provided in Appendix B.

5.2.1 Matrix Valued Sparse Approximation

This subsection shows how the majorization method is used for the first step of the alternating minimization: matrix valued sparse approximation. The updating formula derived here is used in the generalized block relaxation method derived later in this section. For fixed $\mathbf{D}$, we can use the matrix form of the Taylor series inequality (3.10), see Appendix A, to derive the following majorizing function,

$$\begin{aligned} ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 &\leq ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 \\ &\quad + c_X||\mathbf{X} - \mathbf{X}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X} - \mathbf{D}\mathbf{X}^{[n-1]}||_F^2 \\ &= ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \pi_{\mathbf{X}}(\mathbf{X}, \mathbf{X}^{[n-1]}) \end{aligned} \tag{5.4}$$

Algorithm 1 : $\mathcal{SA}(\mathbf{X}_t, \mathbf{D}_t)$

1: **initialization:** $c_X > \|\mathbf{D}_t^T\mathbf{D}_t\|$, $\mathbf{X}^{[0]} = \mathbf{X}_t$
2: **for** $n = 1$ **to** K_X **do**
3: $\quad \mathbf{A} = \frac{1}{c_X}(\mathbf{D}_t^T\mathbf{Y} + (c_X\mathbf{I} - \mathbf{D}_t^T\mathbf{D}_t)\mathbf{X}^{[n-1]})$
4: $\quad \mathbf{X}^{[n]} = \mathcal{S}_\lambda(\mathbf{A})$
5: **end for**
6: **output:** $\mathbf{X}_{t+1} = \mathbf{X}^{[K_X]}$

where $\mathbf{X}^{[n-1]}$ is the coefficient matrix in the previous step, $\pi_{\mathbf{x}}(\mathbf{X}, \mathbf{X}^{[n-1]}) := c_X||\mathbf{X}-\mathbf{X}^{[n-1]}||_F^2 - ||\mathbf{DX} - \mathbf{DX}^{[n-1]}||_F^2$ and $c_X > ||\mathbf{D}^T\mathbf{D}||$ is a constant, where $||.||$ is defined as the spectral norm [HJ85]. This type of majorization has been used for sparse approximation with vector valued coefficients in Chapter 3. $\Phi(\mathbf{D}, \mathbf{X})$ in (5.1) has two terms, $||\mathbf{Y} - \mathbf{DX}||_F^2$ and $\lambda\mathcal{J}_p(\mathbf{X})$. Therefore a function majorizing $\Phi(\mathbf{D}, \mathbf{X})$ is,

$$\Phi(\mathbf{D}, \mathbf{X}) \leq \Phi(\mathbf{D}, \mathbf{X}) + \pi_{\mathbf{x}}(\mathbf{X}, \mathbf{X}^{[n-1]}) \tag{5.5}$$

Let $\mathbf{A} := \frac{1}{c_X}(\mathbf{D}^T\mathbf{Y} + (c_X\mathbf{I} - \mathbf{D}^T\mathbf{D})\mathbf{X}^{[n-1]})$. It can be shown that the optimum of the surrogate objective (5.5), where $p = 1$, is found by shrinking elements in $\mathbf{A}$ [DJ94, DDD04], that is,

$$\{\mathbf{X}^{[n]}\}_{i,j} = \mathcal{S}_\lambda(\mathbf{A}) = \begin{cases} a_{i,j} - \lambda/2\ sign(a_{i,j}) & \lambda/2 < |a_{i,j}| \\ 0 & otherwise. \end{cases} \tag{5.6}$$

$\mathbf{A}$ is the matrix version of Landweber update, which was introduced in (3.16). This iterative update continues until $\mathbf{X}^{[n]}$ converges to the optimum solution. The pseudocode for this coefficient update is presented in Algorithm 1. The operator $\mathcal{S}_\lambda$ is the shrinkage operator defined in (5.6). Note that the optimization problem is column separable, i.e. (5.4) while $\mathbf{D}$ is kept fixed, and it can be solved column by column using a standard iterative shrinkage algorithm, see subsection 3.4.2.1. The formulation of (5.6) is *only* a generalization of the soft shrinkage operator to the matrix space.

5.2.2 Dictionary Update

In the second step of the alternating minimization, we minimize the objective function with respect to $\mathbf{D}$, keeping $\mathbf{X}$ fixed. This constrained minimization problem can be solved using several methods. Among these, fixed-point iteration and iterative gradient projection methods have been suggested for the dictionary updates in [KMR$^+$03, OF97], see also Chapter 4. Here

a majorization minimization technique is used to find the dictionary update.

The quadratic part of the objective function in (5.1) has a bounded curvature when minimizing over $\mathbf{D}$. So again using the Taylor series, the majorizing function is as follows,

$$\begin{aligned} ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 &\leq ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 \\ &\quad + c_D||\mathbf{D}-\mathbf{D}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^{[n-1]}\mathbf{X}||_F^2 \\ &= ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}) \end{aligned} \tag{5.7}$$

where $\mathbf{D}^{[n-1]}$ is the dictionary found in the previous step, $\pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}) := c_D||\mathbf{D}-\mathbf{D}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^{[n-1]}\mathbf{X}||_F^2$ and $c_D > ||\mathbf{X}^T\mathbf{X}||$ is a constant. When $\mathbf{X}$ changes in the sparse approximation step, this spectral norm needs to be re-calculated. The spectral norm of a Hermitian matrix is its largest eigenvalue and various efficient methods have been presented to calculate it [ABB+99].

This majorizing function can be used with different constraints. In the following two subsections, the optimum of (5.7) under bounded Frobenius and column-norm constraints are derived.

5.2.2.1 Constrained Frobenius-Norm Dictionaries

An advantage of using a constraint on the Frobenius-norm of the dictionary is that the learned dictionary can have columns with different norms. Such dictionaries can then be used in the weighted-pursuit framework [DGV06], where atoms with large norms have more chance to appear in the approximations. It has been shown that the average performance of sparse approximation increases when the weights are chosen correctly for the class of signals under study [DGV06].

In the dictionary update step, with the help of a Lagrangian multiplier γ, we turn (5.1) into an unconstrained optimization problem,

$$\min_{\mathbf{D}} \phi_\gamma(\mathbf{D},\mathbf{X}), \tag{5.8}$$

where $\phi_\gamma(\mathbf{D},\mathbf{X})$, for $p=1$, is now defined as,

$$\phi_\gamma(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda\mathcal{J}_1(\mathbf{X}) + \gamma(||\mathbf{D}||_F^2 - c_F). \tag{5.9}$$

Fixing $\mathbf{X}$, the solution of (5.8) is a global minimum if the solution satisfies the K.K.T conditions [Roc70, Theorem 28.1], which are necessary conditions for the optimality of solution. As the admissible set is convex, any minimum of $\phi_\gamma(\mathbf{D}, \mathbf{X})$ is an optimal solution if $\gamma(||\mathbf{D}||_F^2 - c_F) = 0$. Therefore if $||\mathbf{D}||_F^2 \neq c_F$, γ must be zero.

The majorizing function is generated by adding $\pi_{\mathbf{D}}$ to the objective function,

$$\psi_\gamma(\mathbf{D}, \mathbf{D}^{[n-1]}) = \phi_\gamma(\mathbf{D}, \mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D}, \mathbf{D}^{[n-1]}). \tag{5.10}$$

$\mathbf{X}$ has here been omitted from the list of parameters because it is assumed fixed in the dictionary update step. The optimum of this function is at a point with zero gradient,

$$\begin{aligned}\frac{d}{d\mathbf{D}}\psi_\gamma(\mathbf{D}, \mathbf{D}^{[n-1]}) = &-2\mathbf{X}\mathbf{Y}^T + 2\mathbf{X}\mathbf{X}^T\mathbf{D}^{[n-1]^T} + 2c_D\mathbf{D}^T \\ &- 2c_D\mathbf{D}^{[n-1]^T} + 2\gamma\mathbf{D}^T = \mathbf{0}\end{aligned}$$

By solving the above equation we find the optimal dictionary,

$$\mathbf{D}_\gamma^* = \frac{c_D}{\gamma + c_D}\mathbf{B} \tag{5.11}$$

where $\mathbf{B}$ is defined as,

$$\mathbf{B} := \frac{1}{c_D}(\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}\mathbf{X}^T)). \tag{5.12}$$

To satisfy the K.K.T. conditions, a non-negative γ has to be found such that $\gamma(||\mathbf{D}^{[n]}||_F^2 - c_F) = 0$. If $\mathbf{D}_\mathbf{0}^* = \mathbf{B}$ is admissible, we can update the dictionary $\mathbf{D}^{[n]} = \mathbf{B}$. Otherwise we scale $\mathbf{B}$ to have Frobenius-norm equal to $c_F^{1/2}$.

$$\mathbf{D}^{[n]} = \mathcal{P}_{c_F}^F(\mathbf{B}) = \begin{cases} \mathbf{B} & ||\mathbf{B}||_F \leq c_F^{1/2} \\ \frac{c_F^{1/2}}{||\mathbf{B}||_F}\mathbf{B} & otherwise \end{cases} \tag{5.13}$$

The pseudocode for this dictionary update is presented in Algorithm 2. Here $\mathcal{P}$ is the operator $\mathcal{P}_{c_F}^F$ presented in (5.13). In the following, it will be shown that the dictionary updates, subject to the constraints on the column-norms of the dictionaries, have similar algorithms, but with the different operators for $\mathcal{P}$.

If we use an equality in the definition of (5.2), i.e. we demand a *fixed* Frobenius-norm, γ can be-

Algorithm 2 : $\mathcal{DU}(\mathbf{X}_{t+1}, \mathbf{D}_t)$

1: **initialization:** $c_D > \|\mathbf{X}_{t+1}^T \mathbf{X}_{t+1}\|$, $\mathbf{D}^{[0]} = \mathbf{D}_t$
2: **for** $n = 1$ **to** K_D **do**
3: $\quad \mathbf{B} = \frac{1}{c_D}(\mathbf{Y}\mathbf{X}_{t+1}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}_{t+1}\mathbf{X}_{t+1}^T))$
4: $\quad \mathbf{D}^{[n]} = \mathcal{P}(\mathbf{B})$
5: **end for**
6: **output:** $\mathbf{D}_{t+1} = \mathbf{D}^{[K_D]}$

come negative. In this case the decision criteria of (5.13) becomes an equality ($||\mathbf{B}||_F = c_F^{1/2}$). Although it has been guaranteed to not increase the majorizing objective using this update, the solution might not be the global minimum.

5.2.2.2 Constrained Column-Norm Dictionaries

Another often used admissible set in dictionary learning is the set of *fixed* or unit column norm matrices. Instead a bound on the column norms of the dictionary can be used to get a convex admissible set. To make (5.1) an unconstrained optimization problem we need N Lagrangian multipliers (equal to the number of constraints),

$$\min_{\mathbf{D}} \phi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{X}), \tag{5.14}$$

where $\phi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{X})$, for $p = 1$, is now defined as,

$$\phi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + \sum_{i=1}^{N} \gamma_i(\mathbf{d}_i^T \mathbf{d}_i - c_C) \tag{5.15}$$

With this formulation, the K.K.T conditions are,

$$\forall i : 1 \leq i \leq N, \qquad \gamma_i(\mathbf{d}_i^T \mathbf{d}_i - c_C) = 0\ . \tag{5.16}$$

This means that for each i when $\mathbf{d}_i^T \mathbf{d}_i$ is not equal to c_C, γ_i should be zero. (5.14) can be rewritten as

$$\phi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + tr\{\mathbf{\Gamma}(\mathbf{D}^T\mathbf{D} - c_C\mathbf{I})\}, \tag{5.17}$$

where Γ is a diagonal matrix with the γ_i as the i^{th} diagonal element. By adding $\pi_{\mathbf{D}}$, we get the majorizing function,

$$\psi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{D}^{[n-1]}) = \phi_{\mathbf{\Gamma}}(\mathbf{D}, \mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D}, \mathbf{D}^{[n-1]}). \tag{5.18}$$

The gradient is again set to zero and the optimum solution is found to be,

$$\mathbf{D}^*_{\Gamma} = \mathbf{B}(\frac{1}{c_D}\mathbf{\Gamma} + \mathbf{I})^{-1}, \tag{5.19}$$

where $\mathbf{B}$ has the same definition as introduced in (5.12). All γ_i are non-negative and $(\frac{1}{c_D}\mathbf{\Gamma} + \mathbf{I})$ is an (invertible) diagonal matrix. In equation (5.19), by changing γ_i, we multiply the corresponding column of $\mathbf{B}$ by a scalar. We start by setting all $\gamma_i = 0$. For any columns of $\mathbf{D}^*_{\mathbf{0}} = \mathbf{B}$ for which the norm is more than $c_C^{1/2}$, we find the smallest value of γ_i which scales down that column to have the largest acceptable norm ($c_C^{1/2}$).

$$\begin{aligned} \mathbf{D}^{[n]} &= \mathcal{P}^C_{c_C}(\mathbf{B}) = \{\mathbf{b}_j^{[n]}\}_{1 \le j \le N} \\ \mathbf{d}_j^{[n]} &= \begin{cases} \mathbf{b}_j & ||\mathbf{b}_j||_2 \le c_C^{1/2} \\ \frac{c_C^{1/2}}{||\mathbf{b}_j||_2}\mathbf{b}_j & otherwise, \end{cases} \end{aligned} \tag{5.20}$$

where $\mathbf{d}_j$ and $\mathbf{b}_j$ are the j^{th} columns of $\mathbf{D}$ and $\mathbf{B}$ respectively.

Alternatively, we can use a *fixed* column-norm constraint ($\mathscr{D} = \{\mathbf{D}_{d \times N} : ||\mathbf{d}_i||_2 = c_C^{1/2}\}$). Here the algorithm may find a Γ in which some of the γ_i are negative. The dictionary update can then be found by a similar operator as (5.20) but with equality in the decision criteria ($||\mathbf{b}_j||_2 = c_C^{1/2}$) or simply by

$$\mathbf{d}_j^{[n]} = \frac{c_C^{1/2}}{||\mathbf{b}_j||_2}\mathbf{b}_j. \tag{5.21}$$

When the norm of any columns of $\mathbf{B}$ is zero, we have some ambiguity in the update formula. In this case we can shrink the size of the dictionary by deleting this atom or keep the size fixed by introducing a random atom to the dictionary. In practice we have not encountered such an ambiguity.

Algorithm 3 : $\mathcal{DL}(\mathbf{X}_0, \mathbf{D}_0)$

1: **for** $t = 1$ **to** T **do**
2: $\quad \mathbf{X}_{t+1} = \mathcal{SA}(\mathbf{X}_t, \mathbf{D}_t)$
3: $\quad \mathbf{D}_{t+1} = \mathcal{DU}(\mathbf{X}_{t+1}, \mathbf{D}_t)$
4: **end for**
5: **output:** $\mathbf{D}_T$

5.2.3 Generalized block relaxation method for dictionary learning

In the previous subsections a block relaxation method was presented to optimize $\mathbf{X}$ and $\mathbf{D}$ iteratively. In each step, we used an iterative method to find the optimum solution based on one variable while keeping the other variable fixed. The pseudocode for dictionary learning in this framework is presented in Algorithm 3.

Because the joint objective function does not have a fixed bounded curvature, we could not use the majorization method for both parameters jointly. On the other hand, this alternating optimization decreases the rate of convergence as it often oscillates around the optimal path. Instead of fully optimizing with respect to a single parameter in each step, the generalized block relaxation method updates each variable at a time and reduces the objective function, using for example a cyclic selection or any other periodic selection of the parameters. A simple way to choose which parameter to update is to calculate the update based on each parameter and then choose the parameter that decreases the objective function the most. A drawback of this type of parameter selection is that it doubles the computational cost. Another technique is to alternatively update each parameter. For dictionary learning, we found that using more coefficient updates than dictionary updates is in general more beneficial. So one can use p updates of $\mathbf{X}$ followed by q updates of $\mathbf{D}$ where $p \geq q$.

A more complete explanation and a basic convergence proof for the generalized block relaxed dictionary learning algorithm are provided in Appendix B. It is easy to show that the block relaxation method is a special case of the generalized block relaxation method. Therefore convergence of the block relaxation method (alternating minimization) for the dictionary learning follows as a corollary of this result.

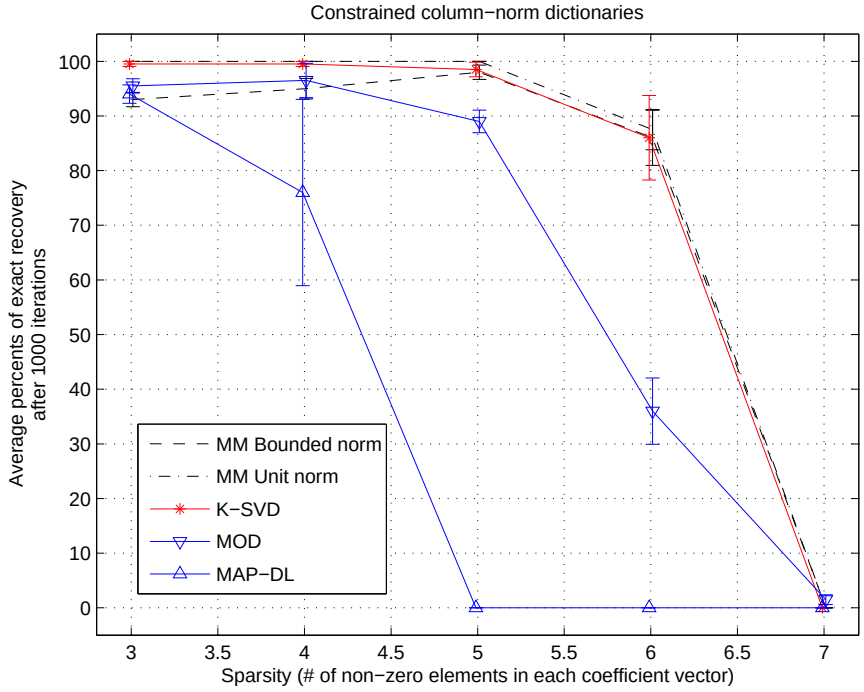

Figure 5.1: *A comparison of the dictionary recovery success rates using different dictionary learning methods under a column-norm constraint.*

5.3 Simulations

The dictionary learning using majorization minimization method is evaluated with synthetic and real data in this section. Using synthetic data with random dictionaries helps us to examine the ability of the proposed methods to recover dictionaries exactly (to within an acceptable squared error). The synthetic data and dictionaries are generated as proposed in [KMR$^+$03] and [AEB06]. To evaluate the performance on real data, audio signals, which have been shown to have some sparse structure, are chosen. The learned dictionary has then been used for audio coding. The Rate-Distortion performances of the sparse coding with the learned and classical dictionaries were thus compared.

5.3.1 Synthetic Data

A 20×40 matrix $\mathbf{D}$ was generated by normalizing a matrix with i.i.d. uniform random entries. The number of non-zero elements in each of the coefficient vectors was selected between 3 and 7. The locations of the non-zero coefficients were selected uniformly at random. A

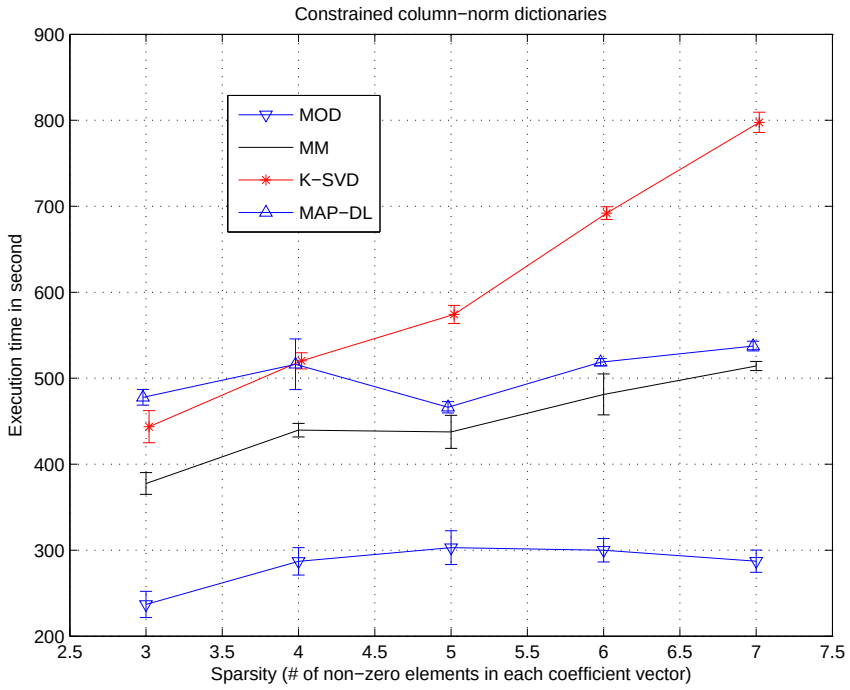

Figure 5.2: *A comparison of the computation costs of the dictionary learning methods under a column-norm constraint.*

set of 1280 training samples was generated where the absolute values of the non-zero coefficients were selected uniformly between 0.2 and 1. In the setting of exact dictionary recovery [KMR+03, AEB06] and under a mild condition, the constrained column-norm dictionary and the K-sparse signals are the global solutions of the dictionary learning problem based on exact sparse representations and the ℓ_1 based exact sparse representation problems, respectively (see for example [GS08]). The proposed algorithm as well as the other dictionary learning algorithms discussed, are proposed for sparse *approximations*, that is, they allow approximation error when calculating the sparse coefficients. To adapt the algorithm to this problem, we assume that the sparse approximation finds the correct support in each step. Once the support has been identified, we can find the best approximation by projecting onto the selected sub-space. This is called debiasing.

The majorization minimization based dictionary learning algorithm is compared to MOD, K-SVD and MAP-DL. The stopping criteria for IT was the distance between two consecutive iterations ($\delta = 3 \times 10^{-4}$) and λ was set to 0.4. The termination conditions for the iterative dictionary learning methods (majorization method for dictionary learning (MM-DL) and MAP-

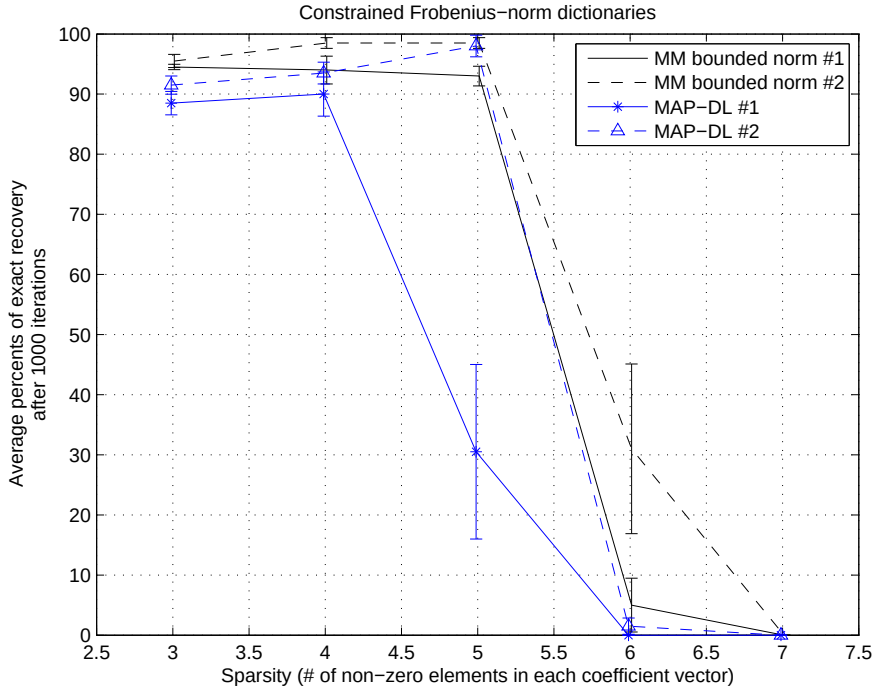

Figure 5.3: *A comparison of the dictionary recovery success rates using MM and MAP dictionary learning methods under a Frobenius norm constraint: 1: Desired dictionary had fixed Frobenius-norm. 2: Desired dictionary had fixed column-norms.*

DL) was set to ($||\mathbf{D}^{[n]} - \mathbf{D}^{[n-1]}||_F \leq 10^{-7}$).

The algorithm was started from a normalized random $\mathbf{D}$ and used 1000 iterations. The learning parameter (γ) in MAP-DL was selected as described in [KMR+03] and γ was down-scaled by a factor of 2^{-j} ($j > 1$) when the algorithm was diverging. To allow a fair comparison, the simulations were repeated for 5 times. If the squared error between a learned and true dictionary element was below 0.01, it was classified as correctly identified. The average percentages and standard deviations are shown in Figure 5.1. It can be seen that in all cases, MM-DL with fixed column-norm and K-SVD recovered nearly the same number of atoms and performed better than the other methods (although, for the signals with less than 6 non-zero coefficients, MM-DL recovered all desired atoms, performance of K-SVD was very close to it). The debiasing process creates some ambiguities in dictionary learning when using the bounded-norm constraints as they reduce the effect of the coefficient magnitudes in the sparsity measure. Therefore, we observe atoms which do not have a boundary norm (here, unit norm), even after 1000 iterations. In this case, we get better results using a fixed column-norm admissible set which resolves this ambiguity. The MAP-DL algorithm did not perform well in this simulation. We

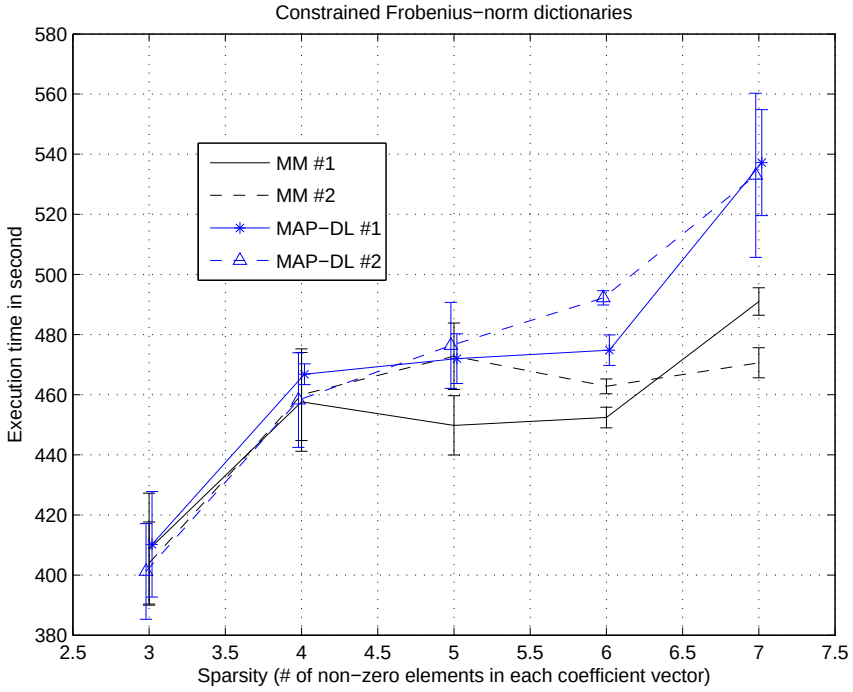

Figure 5.4: *A comparison of the computation costs of the dictionary learning methods under a Frobenius norm constraint.*

guess the reason for this is slow convergence and the use of more iterations might improve the performance.

The computation time of the algorithms are compared in Fig.5.2 for the above simulations. Simulations ran on the Intel Xeon 2.66 GHz dual-core processor machine and both cores were used by Matlab. In this graph the total execution time of the algorithms (sparse approximations plus dictionary updates for 1000 iterations) is shown. MOD was fastest followed by our MM-DL.

We have a larger admissible set when fixing the Frobenius-norm of the dictionary, which makes the problem of exact recovery more complicated and we expect to observe worse performance in terms of exact atom recovery. To test this, we started with a normalized random dictionary, normalized either to have fixed Frobenius-norm or fixed column-norm. The simulations were repeated for 5 trials and the averages and standard deviations of the atom recovery are shown in Fig. 5.3. In these simulations MM-DL performed slightly better than MAP-DL. The other observation in this figure is that when the desired dictionaries have equal column-norms, performance of the algorithms increase but do not reach the performance observed when using the

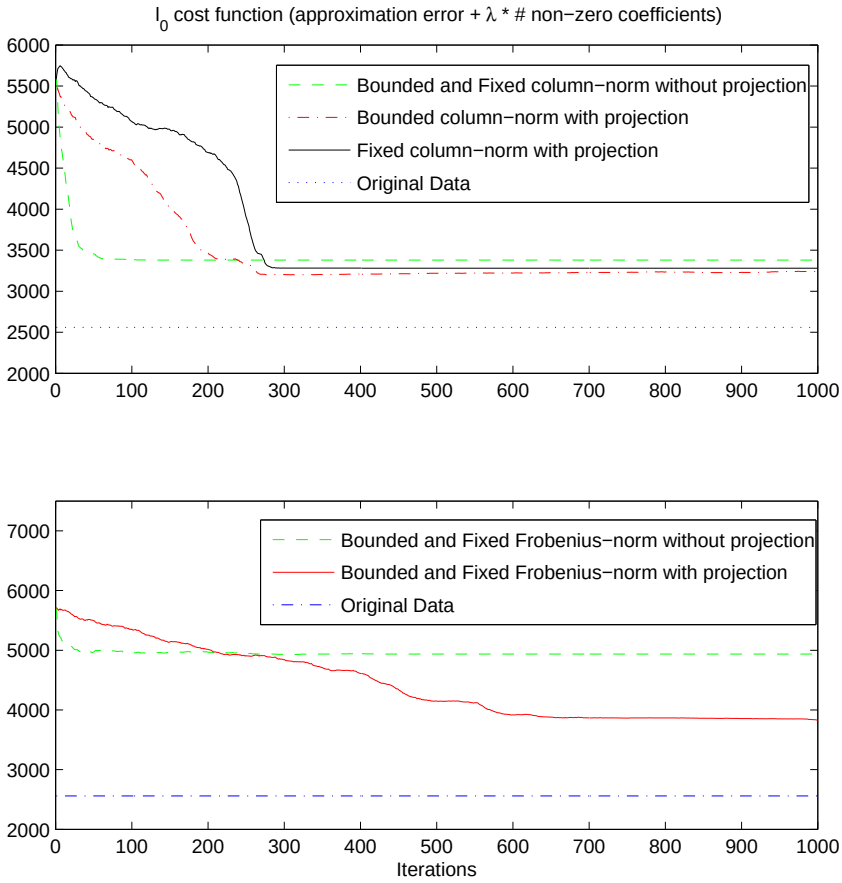

Figure 5.5: *l_0 cost functions of the constrained Frobenius and column -norms dictionary learning algorithms respectively on top and bottom plots.*

more restricted (and appropriate) admissible set. Computation times of the algorithms, on the machine described formerly, are shown in Fig.5.4.

Instead of constraining the dictionaries to have fixed norms, we can use the bounded-norm constraints. To show the possible advantage of these constraints, the simulations are repeated. The results achieved with these constraints are shown in Fig. 5.5. The simulations here are run with and without orthogonal projections on the selected spaces found by sparse approximation method. It can be seen that using bounded-norm admissible set improves performance slightly when constraining the column-norm but it does not change performance of the other method. These plots also show that the orthogonal projection onto the selected spaces can improve overall performances.

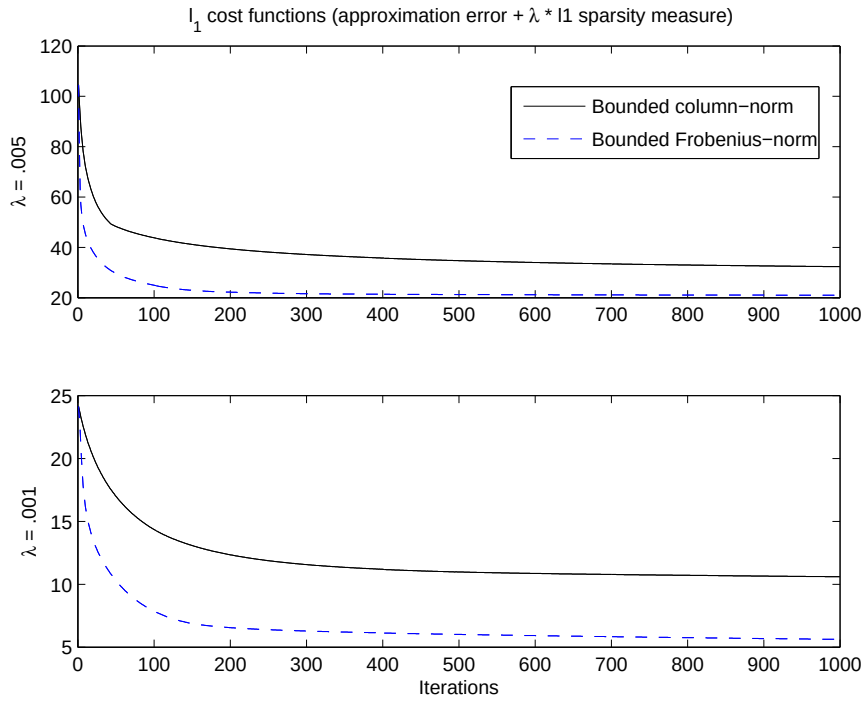

Figure 5.6: *ℓ_1 cost functions for two different Lagrangian multipliers (λ) .005 (top) and .001 (bottom).*

5.3.2 Dictionary Learning for Sparse Audio Coding

In this subsection, the performance of the proposed dictionary learning method is demonstrated on audio signals and it is thus shown that this method is applicable to large dictionary learning problems. An audio sample of more than 8 hours was recorded from BBC radio 3, which plays mostly classical music.

In the first experiment the bounded column-norm and the bounded Frobenius-norm dictionary admissible sets were selected. The audio sample was summed to mono and down-sampled by a factor of 4. From this 12kHz audio signal, 4096 blocks of 256 samples each were randomly selected. The set of dictionaries with the column-norms bounded by c_C is a subset of the set of bounded Frobenius-norm dictionaries, when $c_F = Nc_C$. The dictionary admissible sets with column-norms and Frobenius-norms bounded by $c_C = 1$ and $c_F = N$ were respectively selected. The dictionary was initialized with a 2 times overcomplete random dictionary and the algorithm was ran for 1000 iterations. The objective function against iteration, for two different values of λ, are shown in Fig. 5.6. This figure shows that the optimal bounded Frobenius-norm dictionaries are better solutions for the objective functions.

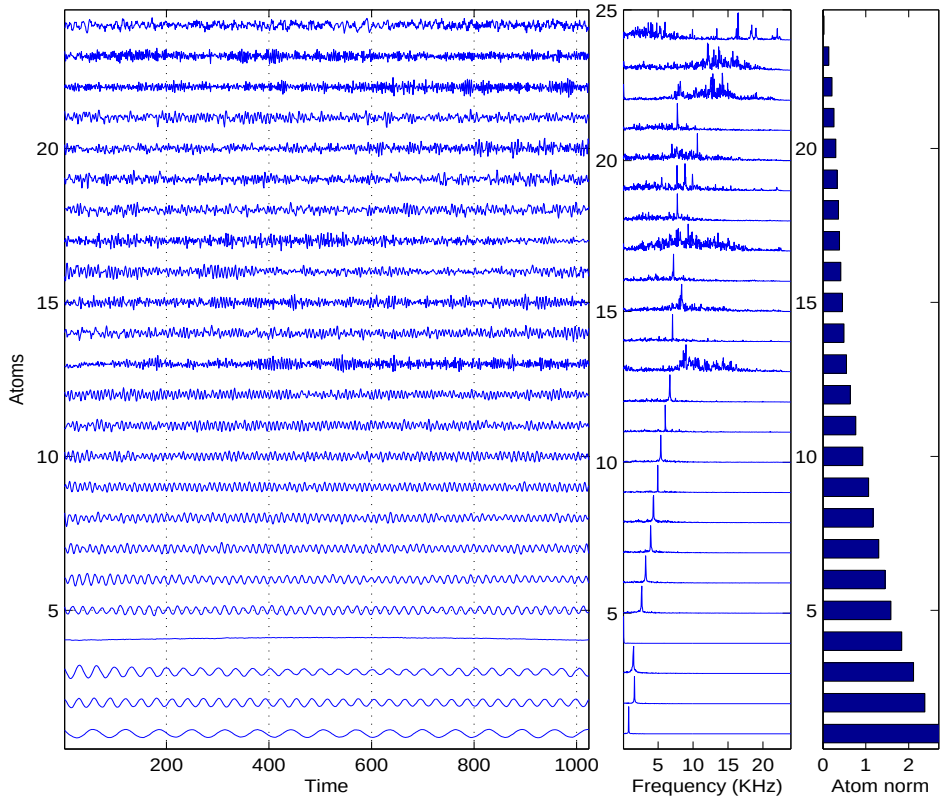

Figure 5.7: *A selection of learned atoms in time (left) and frequency (middle) domain. Their norms are shown in the right panel.*

As a second experiment, an audio coding example is investigated. The dictionary learning method was used with the bounded Frobenius-norm constraint to learn a dictionary based on a training set of 8192 blocks, each 1024 samples long. In this experiment, the aim is to learn the dictionary for a larger block length than the previous experiment. The convergence of the traditional block relaxation method for a problem with this size is very slow. The simulations were thus run with the generalized block relaxation method. Even though the recorded audio had 48k samples per second, the audio had a maximum frequency of 16kHz. Therefore the original audio was downsampled by a factor of 3/2 without any degradation in the audio fidelity. It has been shown that audio can be modeled reasonably well using tonal, transient and noisy residual components [DT02]. A 2 times overcomplete sinusoid dictionary (frequency oversampled DCT) was chosen as the initialization point and the simulations were run with different lambda values for 5000 iterations of alternating optimization of (B.1), which took approximately 8 hours for each λ, running on the machine mentioned in the previous subsection.

A subset of the learned atoms ($\lambda = .01$, $\theta = .01$), which is selected by uniformly sampling the atom indices, is shown in Fig. 5.7. These atoms are shown in the time and frequency domain in the left and middle windows respectively. The norms of the selected atoms are shown in the right window. The number of appearances of each atom, which are sorted based on their ℓ_2 norms, are shown in Fig. 5.8. To design an efficient encoder we only need to use atoms which were used frequently in the representations. Therefore we are able to further shrink the dictionary size. In this test a threshold of 40 appearances (out of 8192) was chosen as the selection criteria. This dictionary was used to find the sparse approximations of 4096 different random blocks, each of 1024 samples, from the same data set. The location (significant bit map) and magnitude of the non-zero coefficients were encoded separately. A uniform scalar quantizer with a double zero bin size was selected to encode the magnitude. The entropy of the coefficients were estimated to approximate the required coding cost. To encode the significant bit map, an i.i.d. distribution was assumed for the location of the non-zero atoms. The same coding strategy was used to code sparse approximations with a two times frequency overcomplete DCT (the initial dictionary used for learning) followed by shrinking based on the number of appearances. For reference, the rate-distortion of the DCT coefficient encoding of the same data was calculated, using the same method of significant bitmap and non-zero coefficients coding. The performance is compared in Fig. 5.9. In the sparse coding methods, the convex hulls of the rate-distortion performances were calculated with different dictionaries, each optimized and shrunk for different bit-rate, are shown in this figure. Using the learned dictionaries for sparse approximation is superior to using the DCT or overcomplete DCT for the range of bit-rates shown.

It would be nice to compare these real data experiments with K-SVD, which is shown to perform well in dictionary learning for medium size problems. However, K-SVD was found to be too slow on problems of this size. For example, one sparse approximations of the signals, using a fast implementation of OMP[2], and one dictionary update approximately took 10 hours and this has to be repeated for a reasonable number of iterations, e.g. 1000 iterations!

5.4 Summary

A new algorithm was presented for dictionary learning and its advantages in different experiments and for different data sets have been shown. The proposed method is very flexible in

[2]Sparsify Toolbox ver. 0.2, `http://www.see.ed.ac.uk/~tblumens/sparsify/sparsify.html`

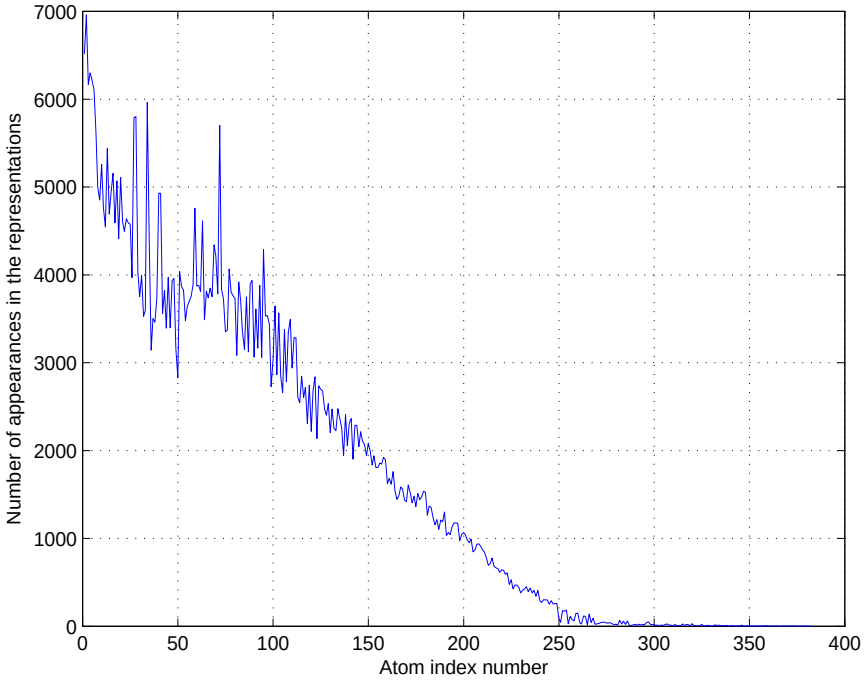

Figure 5.8: *Number of appearances of the learned atoms in the representations of the training samples (of size 8192).*

using different constraints on the dictionaries. Because the problem of dictionary learning was considered in a more general form (bounded norm for dictionaries), better results were possible.

While some of the other methods are based on atom-wise dictionary updates (K-SVD, MAP-DL with unit column-norm prior information), the proposed method updates the whole dictionary at once. Although the computational complexity of each iteration of the given algorithm is roughly cubic, the algorithm was found to be much faster for large scale problems than, for example, K-SVD (which has a higher order of complexity).

The given method solves the dictionary learning problem in a unified framework. This unified framework provides extra flexibility to update the coefficients and the dictionary in a more efficient way. Furthermore, the convergence of the method to a set of fixed points in this framework is proved in Appendix B.

Finally, the constrained Frobenius-norm was shown to increase the performance of dictionary learning by increasing the possible solution set. Audio coding with the learned dictionary showed a superior rate-distortion performance over traditional orthogonal transform coding

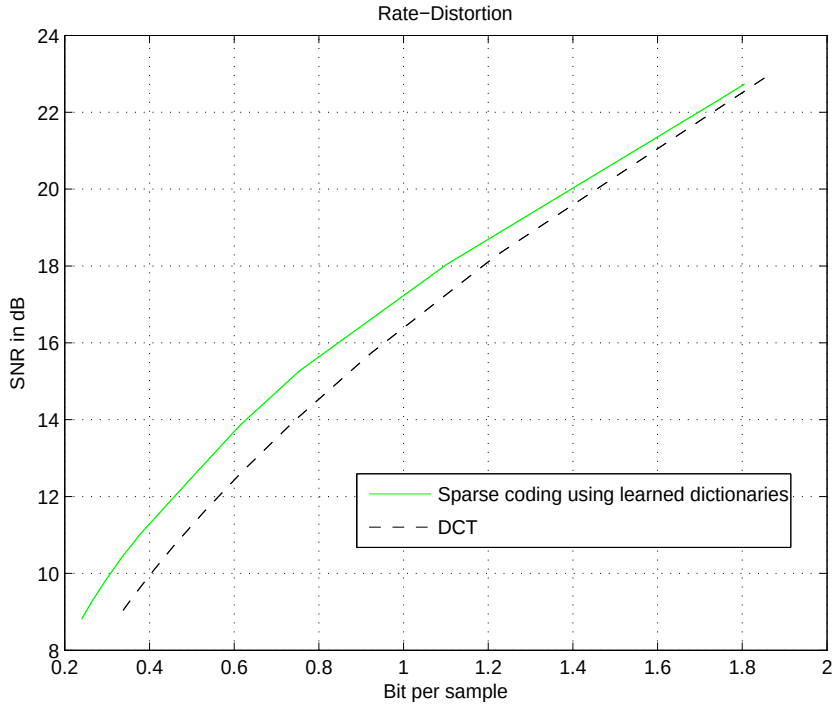

Figure 5.9: *Estimated Rate-Distortion for the audio coding example using the learned dictionary, the shrunk 2 times overcomplete DCT dictionary and the DCT.*

and overcomplete sparse coding with an oversampled DCT.

Chapter 6
Parsimonious Dictionary Learning (PDL)

6.1 Introduction

A very flexible method was introduced in Chapter 5 for dictionary learning under a minimum constraint. It will be shown here that the proposed method can also be used with an extra constraint on the number of atoms.

An application of sparse approximation is sparse coding. The indices of the selected atoms and the associated coefficients are encoded separately in a conventional sparse encoder [FVFK04, DD06, Mal99, RRD08b]. The coding cost of specifying the selected atoms is reduced by reducing the size of dictionary. The minimum size dictionaries are thus more desirable for the coding purpose. When the size of the learned dictionary reduces, the matrix-vector multiplications can also be done faster.

The parsimonious dictionary learning is not only suitable for the coding application but also it can find the dictionary size, when it is unknown. The dictionary size selection is a challenging problem in the sparse approximations. We can increase the sparsity of the approximation by adding more atoms to the dictionary, which increases the dictionary size. In the limit, the dictionary includes all the proposed signals and the signals can therefore be presented by 1-sparse coefficient vectors. It is obvious that finding a sparse representation using such a dictionary and presenting the coefficient vector is very difficult. To reduce the complexities of sparse approximation problem and coefficient representation, we should choose a tractable dictionary size. In practice when the size of the dictionary is unknown, one can start with an oversized dictionary and find the minimum size learned dictionary by gradually decreasing the dictionary size. Finding such a minimal dictionary, given a class of signals, is called “parsimonious dictionary learning”.

A framework for parsimonious dictionary learning will be introduced in this chapter. The problem formulation is followed by a practical algorithm to find an approximate solution. The

proposed framework is shown to give promising results in dictionary recovery. It is also shown that the learned dictionary has advantages over the currently used dictionaries for sparse coding.

6.2 Parsimonious Dictionary Learning Formulation

When the sparsity measure $\mathcal{J}(.)$ is defined as (2.20), it was shown in Chapter 4 that the dictionary learning problem can be formulated as the minimization of a joint objective function based on $\mathbf{D}$ and $\mathbf{X}$, as follows,

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X}} \phi(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D}; \\ &\phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda \mathcal{J}_p(\mathbf{X}), \end{aligned} \tag{6.1}$$

where $\mathcal{J}_p(.)$ is the sparsity measure (2.18). Here p is set to be 1, which makes the minimization over $\mathbf{X}$ convex, if $\mathbf{D}$ is fixed. It was shown that various admissible sets can be used for dictionary learning (e.g. see for example [YBD08]). The bounded column-norm and bounded Frobenius-norm sets have been used as the admissible sets to make the dictionary update a convex problem for a fixed $\mathbf{X}$. The bounded column-norm and Frobenius norm admissible sets are defined in (5.3) and (5.2) respectively.

To find a minimum size dictionary, an additional penalty on the dictionary size can be applied. The new joint optimization problem can be formulated as,

$$\min_{\mathbf{X},\mathbf{D}} \phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D};$$

$$\phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda \mathcal{J}_1(\mathbf{X}) + \theta \| \max_i |\{\mathbf{D}\}_{i,j}| \|_0.$$

Because $||.||_0$ is the operator for counting the number of non-zero elements, it is related to the size of the dictionary, and $\{\mathbf{D}\}_{i,j}$ is the element (i,j) of $\mathbf{D}$. Because $\phi_{\theta,0,\infty}$ is non-convex and non-continuous, we replace the objective function with a relaxed version as follows,

$$\min_{\mathbf{X},\mathbf{D}} \phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D};$$

$$\phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda \mathcal{J}_1(\mathbf{X}) + \theta \mathcal{J}_{1,q}(\mathbf{D}^T) \tag{6.2}$$

where $q \geq 1$ and $\mathcal{J}_{1,q}(.)$ is the joint sparsity measure (2.20). By selecting $q = 1$, the objective function penalizes any non-zero element of the dictionary. With some changes to this frame-

work, it can be used for sparse dictionary learning as introduced in [?]. It is shown in the next chapter that how a variation of formulation (6.2) can be used to find a structured dictionary, which is called compressible dictionary. If $q > 1$, the objective function penalizes the number of atoms more than the sparsity of each atom, which is our aim in the parsimonious dictionary learning. The parameter θ is then the regularization parameter, which controls the sparsity of the dictionary. One can get a smaller dictionary by increasing θ.

This objective function can be minimized in an alternating minimization framework. Although this method is guaranteed to reduce the objective in each iteration, the objective function is not convex as before and has various local minima. The proposed method optimizes $\mathbf{X}$ and $\mathbf{D}$ alternately while keeping the other parameter fixed. In this framework, the non-convex optimization problem is broken into two convex optimization problems, which can be solved using any convex optimization method. Here we again applied the majorization minimization method, see Section 3.4.1.

6.3 PDL with the Majorization Minimization Method

It has been shown in Chapter 5 that the majorization minimization method can be applied to the dictionary learning problem in a block-relaxed framework. It means that in one step we update $\mathbf{X}$ ($\mathbf{X}^{[n]} \rightarrow \mathbf{X}^{[n+1]}$), since $\mathbf{D}$ is kept fixed and in the next step we update $\mathbf{D}$ ($\mathbf{D}^{[n]} \rightarrow \mathbf{D}^{[n+1]}$), since $\mathbf{X}$ is kept fixed. When $\mathbf{D}$ is fixed, $\mathbf{X}^{[n+1]}$ can be found using (5.6). The similarity comes for the fact that since the dictionary is fixed, two optimization problems (5.1) and (6.2), based on $\mathbf{X}$, are equivalent. The difference of PDL and DLMM is only in the dictionary update formula which will be found for PDL in the next section.

6.3.1 PDL: Dictionary Update Step

The objective function $\phi_{\theta,1,q}$ is convex when $\mathbf{X}$ is fixed. For fixed $\mathbf{X}$, the joint sparsity penalty is decoupled by adding $\pi_{\mathbf{D}}$ to the objective function,

$$\psi_{\theta,1,q}(\mathbf{D}, \mathbf{D}^{[n-1]}) = \phi_{\theta,1,q}(\mathbf{D}, \mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D}, \mathbf{D}^{[n-1]}). \tag{6.3}$$

By separating the terms depending on $\mathbf{D}$, the surrogate cost can be written as,

$$\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) \propto c_s tr\{\mathbf{D}\mathbf{D}^T - 2\mathbf{B}\mathbf{D}^T\} + \mathcal{J}_{1,q}(\mathbf{D}^T) \tag{6.4}$$

where $\mathbf{B} = \frac{1}{c_D}(\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}\mathbf{X}^T))$, which was defined in (5.12). The dictionary constraint is introduced into the objective function using Lagrangian multipliers. Let $\mathbf{d}_j$ and $\mathbf{b}_j$ be the j^{th} columns of $\mathbf{D}$ and $\mathbf{B}$ respectively. The objective function, using the bounded column-norm (5.3), can be written as,

$$\begin{aligned}
\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) &\propto \sum_j (tr\{\tau_j^2\mathbf{d}_j\mathbf{d}_j^T - 2\mathbf{b}_j\mathbf{d}_j^T\} + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q) \\
&= \sum_j (\tau_j^2\mathbf{d}_j^T\mathbf{d}_j - 2\mathbf{d}_j^T\mathbf{b}_j + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q) \\
&\propto \sum_j ((\tau_j\mathbf{d}_j - \mathbf{b}_j/\tau_j)^2 + \tfrac{\theta}{c_D\tau_j}||\tau_j\mathbf{d}_j||_q) \\
&= \sum_j \psi_q^{\frac{\theta}{c_D\tau_j}}(\tau_j\mathbf{d}_j, \mathbf{b}_j/\tau_j)
\end{aligned} \tag{6.5}$$

where $\psi_q^\alpha(\mathbf{v},\mathbf{w}) = (\mathbf{w}-\mathbf{v})^2 + \alpha||\mathbf{v}||_q$, $\tau_j = (1+\gamma_j/c_D)^{1/2}$ and γ_j are the Lagrangian multipliers. To minimize (6.5), we can minimize the first term by minimizing ψ_q^α for each $\mathbf{d}_j$ independently. With the help of two lemmas presented in [FR08b], we can find the optimum of ψ_q^α based on $\mathbf{d_j}$ for $q = 1, 2$ and ∞. The minimum of $\psi_q^\alpha(\mathbf{v},\mathbf{w})$ based on $\mathbf{v}$ [FR08b, Lemma 4.1] is,

$$\min_{\mathbf{v}} \psi_q^\alpha(\mathbf{v},\mathbf{w}) = \mathbf{w} - \mathcal{P}_\alpha^{q'}(\mathbf{w}) \tag{6.6}$$

where $\mathcal{P}_\alpha^{q'}$ is the orthogonal projection onto the dual norm ball with radius $\mathbf{w}$ and the dual norm is defined as $||.||_{q'}$ with $1/q' + 1/q = 1$. This minimization problem can be solved analytically for some q [FR08b, Lemma 4.2]. Here we derive the dictionary update formula for $q = 2$.

$$\begin{aligned}
\mathbf{b}_j^* &= \arg\min_{\mathbf{d}_j} \psi_2^{\frac{\theta}{c_s\tau_j}}(\tau_j\mathbf{d}_j, \mathbf{b}_j/\tau_j) \\
&= \begin{cases} \frac{1}{\tau_j^2}(1 - \frac{\theta}{2c_D||\mathbf{b}_j||_2})\,\mathbf{b}_j & \frac{\theta}{2c_D} < ||\mathbf{b}_j||_2 \\ 0 & otherwise\ . \end{cases}
\end{aligned} \tag{6.7}$$

When all γ_j are non-negative, for any inadmissible $\mathbf{b}_j^*$ with $\tau_j = 1$ ($\gamma_j = 0$), one can decrease $||\mathbf{d}_j^*||_2$ to $c_c^{1/2}$ by increasing τ_j to satisfy the K.K.T conditions. The dictionary update is

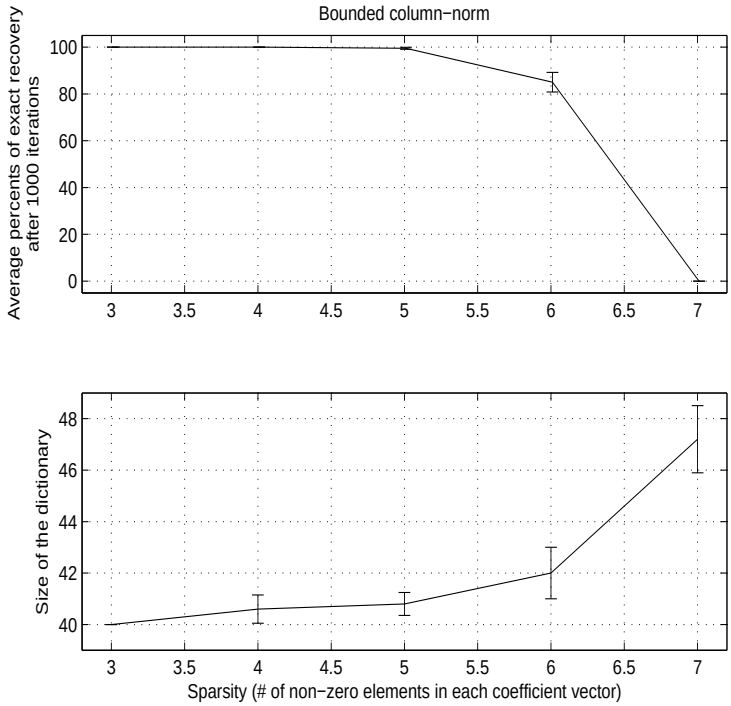

Figure 6.1: *Exact recovery with the constrained column-norm.*

therefore done by calculating $\mathbf{B}$ followed by (6.7) ($\tau_j = 1$) and orthogonal projection onto the convex set (5.3).

When we are looking for a bounded Frobenius-norm dictionary, the dictionary update can be derived using a similar approach, using orthogonal projection onto (5.2) instead of (5.3).

6.4 Simulation

The proposed method is evaluated with synthetic and real data. Using synthetic data with random dictionaries helps us to examine the ability of the proposed methods to recover dictionaries exactly (to within an acceptable squared error). To evaluate the performance on real data, a set of audio signals were chosen. The learned dictionary is applied to the audio coding problem to show improvements in Rate-Distortion performance, in comparison with coding using classical dictionaries.

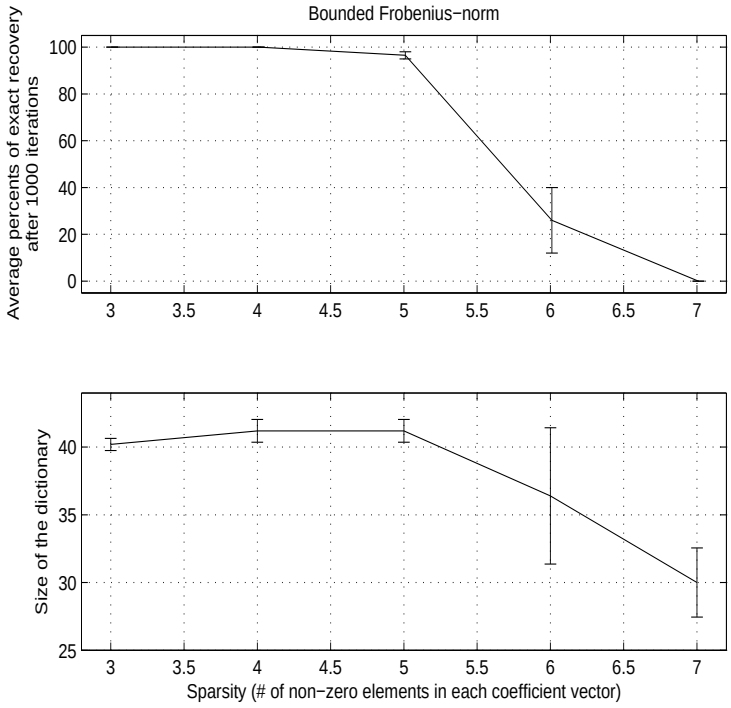

Figure 6.2: *Exact recovery with the bounded Frobenius column-norm.*

6.4.1 Synthetic Data

A 20×40 matrix $\mathbf{D}$ was generated by normalizing a matrix with i.i.d. uniform random entries. The number of non-zero elements in each of the coefficient vectors was selected between 3 and 7. The locations of the non-zero coefficients were selected uniformly at random. We generated 1280 training samples where the absolute values of the non-zero coefficients were selected uniformly between 0.2 and 1. We debiased all the sparse approximations by orthogonally projecting onto the space spanned by atoms with non-zero coefficients.

It is assumed that the desired dictionary size is unknown but bounded. The simulations were started with four times overcompete dictionaries (two times larger than the desired dictionary size). The dictionary updates were based on the joint sparsity objective function (6.2) (with $\theta = 0.05$, $p = 1$ and $q = 2$). The average percentages of exact atom recovery, i.e. absolute inner product of the learned atom with one of the atoms in the original dictionary is more than 0.99, for 5 trials are shown in Fig. 6.1 and 6.2. The percentages of the exact recovery of the original atoms, regardless of the learned dictionary size, is plotted in these figures. The size of dictionaries after 1000 iterations are shown in the lower plots . With this θ we identified the

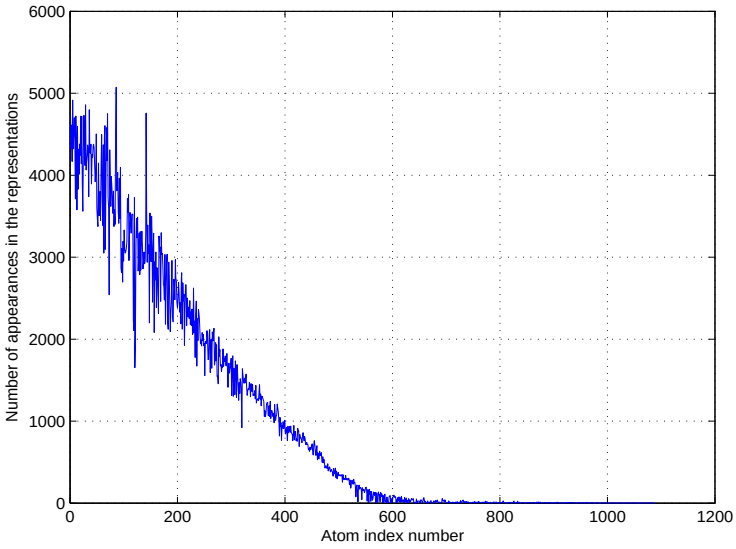

Figure 6.3: *Number of appearances in the representations of the training blocks (of size 8192).*

size correctly but for less sparse signals (higher k), less accurate results are yielded.

6.4.2 Parsimonious Dictionary Learning for Sparse Audio Coding

The performance of the proposed dictionary learning method on audio signals is demonstrated in this section. An audio sample of more than 8 hours was recorded from BBC radio 3, which plays mostly classical music. The proposed method with the bounded Frobenius-norm constraint was used to learn a dictionary based on a training set of 8192 blocks, each 1024 samples long.

In this experiment, instead of fully optimizing over one parameter ($\mathbf{X}$ or $\mathbf{D}$) before switching to the other one, each parameter was updated for a small number of iterations and then switched to the other one. This type of alterante optimization was found to be faster in practice.

A 2 times overcomplete sinusoid dictionary (frequency oversampled DCT) was selected as the initialization point and the simulations were run with different lambda values for 5000 iterations of alternating optimization of (6.5). The number of appearances of each atom, which are sorted based on their ℓ_2 norms, are shown in Fig. 6.3. The atoms that were often used in the representations have been used to design an efficient encoder. Therefore the dictionary size were be able to further shrink. In this test a threshold of 40 appearances (out of 8192) has been

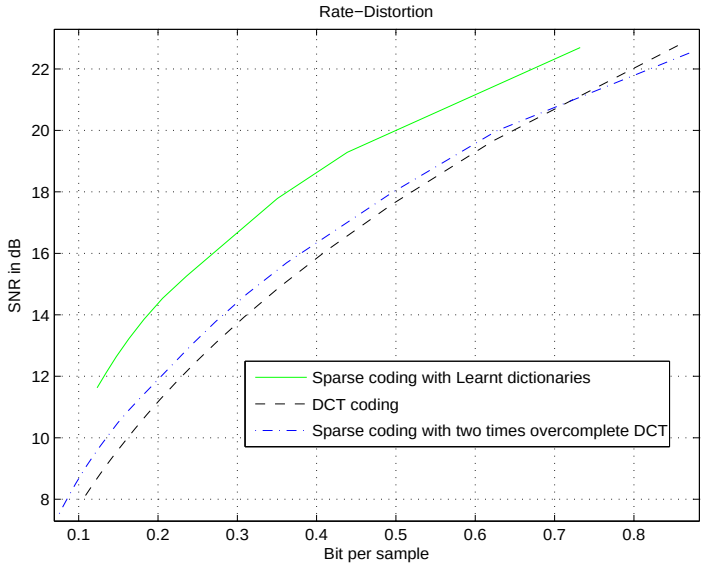

Figure 6.4: *Estimated Rate-Distortion for the audio coding.*

chosen as the selection criteria. This dictionary was used to find the sparse approximations of 4096 different random blocks, each of 1024 samples, from the same data set. The location (significant bit map) and magnitude of the non-zero coefficients were encoded separately. A uniform scalar quantizer with a double zero bin size was used here to code the magnitude. The entropy of the coefficients was estimated to approximate the required coding cost. To encode the significant bit map, an i.i.d. distribution was assumed for the location of the non-zero atoms. The same coding strategy was used to code sparse approximations with a two times frequency overcomplete DCT (the initial dictionary used for learning) followed by shrinking based on the number of appearances. For reference the rate-distortion of the DCT coefficient encoding of the same data was calculated using the same method of significant bitmap and non-zero coefficients coding. The performance is compared in Fig. 6.4. In the sparse coding methods, the convex hulls of the rate-distortion performances calculated with different dictionaries, each optimized and shrunk for different bit-rates, are shown in this figure. Using the learned dictionaries for sparse approximation is superior to using the DCT or overcomplete DCT for the range of bit-rates shown.

6.5 Conclusions

A formulation for the parsimonious dictionary learning was introduced and it has been shown how we can solve the dictionary learning problem approximately, by imposing a penalty on the size of the dictionary, using a majorization minimization method. By a set of simulations, it was shown that the algorithm often recovers a dictionary with the correct size, when the signal is highly sparse. The learned dictionary was then used for sparse coding. The advantages, particularly at low bit-rates, over a standard overcomplete and an orthogonal dictionaries were shown. Although the results are promising, more investigations are needed to find a relevant parameter θ.

Chapter 7
Compressible Dictionary Learning (CDL)

7.1 Introduction

In the previous chapters it was explained that an initial dictionary can be adapted to a set of training samples. An important disadvantage of using learned dictionaries is the lack of structures that would allow fast implementations. In this chapter, we propose a new dictionary model. The imposed model is not only flexible enough to allow the dictionary to adapt to the given class of signals, but also allows the learned dictionary to be implemented efficiently. The dictionary learning, with a compressibility assumption on the dictionary, is a well-defined non-convex optimization problem. We present a practical algorithm to approximately solve compressible dictionary learning. This algorithm can be shown to converge to a local minimum or a set of local minima, by using a similar result as that used for dictionary learning with the majorization minimization method in Appendix B.

7.2 Compressible Dictionary

To propose the compressible dictionary model, we need to introduce the concept of signal compressability [CRT06b]. A signal ψ is defined to be compressible when the entries obey a power law,

$$|\psi|_{(k)} \leq c_r k^{-r}, \tag{7.1}$$

where $|\psi|_{(k)}$ is the k^{th} largest value of ψ, $r \geq 1$ and c_r is a constant[1]. In a similar way we call a matrix $\mathbf{\Psi}$ to be compressible if its entries obey a power law. For example, it has been shown that wavelet coefficients of a piecewise smooth image is a compressible matrix. An important feature of this class of signals, which has been used in the compressed sensing

[1]Such a model was initially introduced in "Nonlinear Approximation" theory, see for example [DeV98], to specify the accuracy of k terms approximations. In this context, a signal ψ is in the approximation space $\mathcal{A}^r$ if it followes the model (7.1)

[CRT06a, Don06], is that a K-sparse signal approximates a compressible signal with a good approximation. Let $\mathbf{\Psi}_K$ be the matrix with the K largest elements of $\mathbf{\Psi}$, and let the other elements be zero. $\mathbf{\Psi}_K$ is the best estimate for $\mathbf{\Psi}$, in the Hilbert space equipped with the inner-product $\langle \mathbf{A}, \mathbf{B} \rangle := \mathrm{tr}\{\mathbf{A}^T\mathbf{B}\}$, and the approximation error is upper-bounded by the following formula,

$$\|\mathbf{\Psi} - \mathbf{\Psi}_K\|_F \leq c'_r K^{-r+1/2}. \tag{7.2}$$

This property has been used in compressed sensing of the compressible signals by recovering the best K-sparse signals which are good approximations of the compressed signals [CRT06b].

***Definition* 7.2.1.** A dictionary $\mathbf{D} \in \mathbb{R}^{d\times N}$ is called *compressible* when for a given full-rank matrix $\mathbf{\Phi} \in \mathbb{R}^{d\times M}$, called mother dictionary, $\mathbf{D}$ could be generated using the following linear model,

$$\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi}, \tag{7.3}$$

where $\mathbf{\Psi} \in \mathbb{R}^{M\times N}$ is a compressible matrix and $M \geq d$.

The compressible dictionaries have two important features:

- *Complexity of Approximation*: It was shown in (7.2) that a K-sparse matrix $\mathbf{\Psi}_K$ can approximate $\mathbf{\Psi}$ with a good approximation. To approximate a compressible dictionary, given $\mathbf{\Phi}$, one can find the best K-sparse $\mathbf{\Psi}_K$. The approximation complexity of $\mathbf{D}$ reduces from $d \times N$ to K as a result.

  ***Proposition* 7.2.1.** Let $\mathbf{D}$ be a compressible dictionary with the generative model (7.3) and $|\psi|_{(k)} \leq c_r k^{-r}$. The approximation error of the generated K-sparse dictionary $\mathbf{D}_K = \mathbf{\Psi}\mathbf{\Psi}_K$ decays rapidly by increasing K. The upper-bound of approximation error is as follows,

  $$\|\mathbf{D} - \mathbf{D}_K\|_F \leq c'_r \|\mathbf{\Phi}\| K^{-r+1/2}, \tag{7.4}$$

  where $\|\mathbf{\Phi}\|$ is the operator norm [GVL96] of $\mathbf{\Phi}$ and c'_r is a constant defined in (7.2).

  Proof. The proof is based on the definition of the operator norm. For a bounded operator $\mathbf{\Phi} : \mathbb{R}^{N\times L} \to \mathbb{R}^{d\times L}$ in a matrix Hilbert space, the operator norm is defined by,

  $$\|\mathbf{\Phi}\| = \max_{\mathbf{\Theta}\in\mathbb{R}^{N\times L}} \frac{\|\mathbf{\Phi}\mathbf{\Theta}\|_F}{\|\mathbf{\Theta}\|_F}. \tag{7.5}$$

It provides the following inequality for all $\mathbf{\Theta} \in \mathbb{R}^{N \times L}$,

$$\|\mathbf{\Phi\Theta}\|_F \leq \|\mathbf{\Phi}\|.\|\mathbf{\Theta}\|_F. \tag{7.6}$$

If $\mathbf{\Theta} = \mathbf{\Psi} - \mathbf{\Psi}_K$, then,

$$\begin{aligned} \|\mathbf{D} - \mathbf{D}_K\|_F &\leq \|\mathbf{\Phi}\|.\|\mathbf{\Psi} - \mathbf{\Psi}_K\|_F \\ &\leq c'_r \|\mathbf{\Phi}\| K^{-r+1/2}, \end{aligned} \tag{7.7}$$

where the second inequality is due to the inequality (7.4). □

Note that the operator norm of $\mathbf{D}_\Delta := \mathbf{D} - \mathbf{D}_K$ is upper-bounded by $\|\mathbf{D} - \mathbf{D}_K\|_F$. Therefore the error caused by the operator $\mathbf{D}_\Delta$ also tends to zero, when $K \to d.N$ at least with the decay rate presented in Proposition 7.2.1.

- *Fast multiplication*: Any vector multiplication with $\mathbf{D}$ can be done in two steps, a multiplication with the sparse matrix $\mathbf{\Psi}_K$ followed by a multiplication with $\mathbf{\Phi}$. Multiplication with the sparse matrix $\mathbf{\Psi}_K$ is $\mathscr{O}(K)$. When $\mathbf{\Phi}$ has structures which provide fast matrix-vector multiplication, e.g. Fourier and wavelets, the matrix multiplication can be done in $\mathscr{O}(N \log N)$ or better. In the practical applications $K \ll N \log N$. Therefore the overall complexity of multiplication with $\mathbf{D}$ is reduced to $N \log N$. It is a significant improvement over the traditional non-structured dictionary multiplication, for example found by dictionary learning, where complexity is $d\ N$.

Dictionary approximation using the generative model (7.3) was introduced in [RBC98], and formulated later in [NZ02], where $\mathbf{\Psi}$ is column-wise k-sparse. If this approximation is accurate, the sparse approximation will be accelerated significantly, while the sparsity of the approximation stays almost the same. A disadvantage of such dictionary approximation is that $\mathbf{D}$ and $\mathbf{\Phi}$ are fixed and the k-term approximation might not be very accurate, e.g. $\|\mathbf{D} - \mathbf{\Phi\Psi}_K\|_F$ is large. The contribution of this chapter is to let $\mathbf{D}$ be variable and we find a compressible $\mathbf{\Psi}$, which has a better coefficient decay rate. Such a compressible $\mathbf{\Psi}$ provides a more accurate k-term approximation.

In the next section a formulation for the dictionary learning under compressability condition of the dictionary is presented. In this framework a practical algorithm is presented which can find an approximate solution for the dictionary learning problem. In section 7.5 the advantages of

proposed framework will be demonstrated by an experiment.

7.3 Problem Formulation

Let a set of training samples $\{\mathbf{y}_l \in \mathbb{R}^d\}_{l\in\mathbb{L}}$, which builds the matrix of training samples $\mathbf{Y} \in \mathbb{R}^{d\times L=|\mathbb{L}|}$, and the mother dictionary $\mathbf{\Phi} \in \mathbb{R}^{d\times N}$ be given. In the dictionary learning problem, the sparse approximation $\mathbf{X}$ and the dictionary generator matrix $\mathbf{\Psi}$ are unknown.

In a standard sparse signal recovery, $\mathbf{\Psi}$ is known and the denoised $\mathbf{X}$ found, for example, by solving BPDN problem (2.8). The denoised $\mathbf{\Psi}$ could be found using a similar method, when $\mathbf{X}$ is given. In the dictionary learning problem, where $\mathbf{\Psi}$ and $\mathbf{X}$ are both unknown, one can minimize a joint objective to find these parameters. Therefore the problem can be formulated as a non-convex optimization problem as follows,

$$\min_{\mathbf{\Psi},\mathbf{X}} \nu(\mathbf{\Psi},\mathbf{X}) \ : \ \nu(\mathbf{\Psi},\mathbf{X}) = \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda\mathcal{J}_p(\mathbf{X}) + \gamma\mathcal{J}_q(\mathbf{\Psi}), \tag{7.8}$$

where $\mathcal{J}_{\mathcal{P}}(.)$ for $\mathcal{P} = \{p, q\}$ is the sparsity measures previously defined in (2.18) and $\lambda, \gamma \in \mathbb{R}^+$. Let $p = q = 1$. The sparsity measures $\mathcal{J}_p(.)$ and $\mathcal{J}_q(.)$ are now ℓ_1 norms, which turns (7.8) into be a *bi-convex* optimization problem. The parameters λ and γ control the sparsity of the coefficients and the dictionary generator matrix.

It can easily be shown that (7.8) is a well-defined optimization problem using the following lemma.

***Lemma* 7.3.1.** *The solution set of the problem (7.8) is bounded.*

Proof. ν is a continuous function. Let epigraph of ν at $(\mathbf{\Psi},\mathbf{X})$ be $\mathrm{epi}(\nu,(\mathbf{\Psi},\mathbf{X}))$. $\mathrm{epi}(\nu,(\mathbf{0},\mathbf{0}))$ for a continuous function ν is compact [BV04]. The solution set is bounded because it is a subset of $\mathrm{epi}(\nu,(\mathbf{0},\mathbf{0}))$. □

The scale ambiguity in the standard dictionary learning is often resolved by inducing a constraint on the dictionary. Although the formulation (7.8) does not have scale ambiguity, it might have non-unique solutions, which are not the column permutated of each other.

***Remark* 7.3.1.** Let $(\mathbf{\Psi}^*,\mathbf{X}^*)$ be a non-zero solution of (7.8) and $\alpha := \gamma J_{1,1}(\mathbf{\Psi}^*)/\lambda J_{1,1}(\mathbf{X}^*)$. If $\alpha \neq 1$ then $(\frac{1}{\alpha}\mathbf{\Psi}^*, \alpha\mathbf{X}^*)$ is another solution of (7.8).

7.4 CDL Algorithm

The problem proposed in Section 7.3 is non-convex and non-differentiable. The difficulty of the problem can be reduce with the block-relaxation method which has been used for standard dictionary learning and introduced in Chapter 4. In this framework, we minimize $\nu(\mathbf{\Psi}, \mathbf{X})$ with respect to $\mathbf{\Psi}$ or $\mathbf{X}$ each time, when the other is fixed. In other word, by starting from an initial solution $(\mathbf{\Psi}^{[0]}, \mathbf{X}^{[0]})$, the algorithm refines the solution by $\mathbf{\Psi}^{[n]} \rightarrow \mathbf{\Psi}^{[n+1]}$ or $\mathbf{X}^{[n]} \rightarrow \mathbf{X}^{[n+1]}$ to reduce $\nu(\mathbf{\Psi}, \mathbf{X})$. When we reduce such a positive objective at each step, the algorithm is stable due to the Lyapunov's second theorem. Because $\nu(\mathbf{\Psi}, \mathbf{X})$ is continuous, the convergence of the algorithm, to a set of fixed points, can easily be shown using a statement similar to Appendix B.

In this thesis, p and q are chosen to be 1. Therefore the optimization problem (7.8) is bi-convex and each step of block-relaxed minimization can be done by any convex optimization method in theory. A method is suitable for the dictionary learning problem if it can handle a large scale problem. The majorization minimization method, which was also used for sparse approximation in Chapter 3 and standard dictionary learning in Chapter 5, has been chosen here to optimize ν with respect to each parameter. Because this method is parallelizable and only needs matrix-matrix multiplications, it is applicable to the large size optimization problems like dictionary learning. This method has been described in Chapter 3 in detail.

The majorizing functions for ν, by fixing either $\mathbf{X}$ or $\mathbf{\Psi}$, are presented in the next subsection. The majorized objective should be minimized in order to find the update of each parameter. The majorizing objective is convex with respect to corresponding parameters. The update formula will then be derived, by letting zero to be in the subgradient of objective.

7.4.1 Derivation of the Majorizing Functions for CDL

The objective ν is an additive combination of the quadratic part $\|\mathbf{\Phi\Psi X} - \mathbf{Y}\|_F^2$ and the sparsity measures. The quadratic part of the objective ν has bounded curvature when one parameter is fixed. A majorizing function could be generated using the Taylor series in matrix form, see Appendix A. This operation can simply be done by adding an appropriate strictly convex function to ν [DDD04], see Section as an example.

Two distinctive majorizing functions are derived for updating $\mathbf{X}$ and $\mathbf{\Psi}$, for fixed $\mathbf{\Psi}$ and $\mathbf{X}$

respectively. These are followed by deriving the update formulas for each case.

- *Update formula for* $\mathbf{X}$*:*
  Let $\nu_{\mathbf{\Psi}}(\mathbf{X}) : \mathbb{R}^{N\times L} \rightarrow \mathbb{R}^{+}$ be $\nu(\mathbf{\Psi}, \mathbf{X})$ at a fixed $\mathbf{\Psi}$. The majorizing function is found by adding $\nu_{\mathbf{\Psi}}(\mathbf{X})$ and $\pi_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]})$, which is found by,

$$\pi_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]}) = c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \|\mathbf{X} - \mathbf{X}^{[n]}\|_F^2 - \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{X} - \mathbf{\Phi}\mathbf{\Psi}\mathbf{X}^{[n]}\|_F^2, \tag{7.9}$$

  where $c_{\mathbf{\Phi}} > \|\mathbf{\Phi}^T\mathbf{\Phi}\|$ and $c_{\mathbf{\Psi}} > \|\mathbf{\Psi}^T\mathbf{\Psi}\|$. The majorizing objective $\mu_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]})$ is then found by,

$$\begin{aligned}\mu_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]}) = tr\{c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}\mathbf{X}^{[n]}) \\ + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]})\} + \lambda J_{1,1}(\mathbf{X}) + c_{\mathbf{X}},\end{aligned} \tag{7.10}$$

  where $c_{\mathbf{X}}$ is a constant with respect to $\mathbf{X}$. $\mu_{\mathbf{\Psi}}$ is a non-differentiable convex function. The matrix $\mathbf{0}$ is then in the subgradient of $\mu_{\mathbf{\Psi}}$ at the minimum. We know that $\mathbf{X}^{[n+1]} = \arg\min_{\mathbf{X}} \mu(\mathbf{X}, \mathbf{X}^{[n]})$. Therefore $\mathbf{X}^{[n+1]}$ should satisfy,

$$0 \in \partial\mu_{\Psi}(\mathbf{X}^{[n+1]}, \mathbf{X}^{[n]}), \tag{7.11}$$

  where,

$$\begin{aligned}\partial\mu_{\Psi}(\mathbf{X}, \mathbf{X}^{[n]}) = 2c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X} - 2(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}\mathbf{X}^{[n]}) \\ + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]}) + \lambda\partial J_{1,1}(\mathbf{X}).\end{aligned} \tag{7.12}$$

  $\mathbf{X}^{[n+1]}$ can easily be found using the soft-shrinkage operator, which was introduced in Chapter 3 to find sparse approximations using the iterative thresholding method.

$$\mathbf{X}^{[n+1]} = \mathcal{S}_{\lambda/2}\left[\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{\Psi}}}(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}\mathbf{X}^{[n]}) + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]})\right] \tag{7.13}$$

- *Update formula for* $\mathbf{\Phi}$*:*
  Note there are similarities among the formulas for sparse approximation update (3.17), dictionary learning updates (5.13) and (5.20) and CDL sparse coefficient matrix update (7.13). This is caused by the structural similarities in the objective which is the left-hand or right-hand multiplication of the optimizing parameter with other terms in the quadratic component of the objective. Here, the optimization parameter is multiplied

from right and left-hand side simultaneously, in the quadratic term. Therefore we expect to derive a slightly different updating formula.

Let $\nu_{\mathbf{X}}(\mathbf{\Psi}) : \mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{+}$ be $\nu(\mathbf{\Psi}, \mathbf{X})$ at a fixed $\mathbf{X}$. A similar technique to the previous part can be used to generate the majorizing function for $\nu_{\mathbf{X}}(\mathbf{\Psi})$. Here, $\pi_X(\mathbf{\Psi}, \mathbf{\Psi}^T)$ is calculated by,

$$\pi_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]}) = c_{\mathbf{\Phi}} c_{\mathbf{X}} \|\mathbf{\Psi} - \mathbf{\Psi}^{[n]}\|_F^2 - \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{X} - \mathbf{\Phi}\mathbf{\Psi}^{[n]}\mathbf{X}\|_F^2. \tag{7.14}$$

where $c_{\mathbf{X}} > \|\mathbf{X}\mathbf{X}^T\|$. The majorizing objective $\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]})$ is now found to be,

$$\begin{aligned}\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]}) = tr\{c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^T\mathbf{\Psi} - 2\mathbf{\Psi}^T(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}^{[n]}\mathbf{X})\mathbf{X}^T \\ + c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^{[n]})\} + \lambda J_{1,1}(\Psi) + c_{\Psi},\end{aligned} \tag{7.15}$$

where c_{Ψ} is a constant with respect to $\mathbf{\Psi}$. The matrix $\mathbf{0}$ should be in the subgradient of $\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]})$ at the minimum $\mathbf{\Psi}^* = \mathbf{\Psi}^{[n+1]}$. This provides the following update formula,

$$\mathbf{\Psi}^{[n+1]} = \mathcal{S}_{\gamma/2}\,[\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{X}}}(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}^{[n]}\mathbf{X})\mathbf{X}^T + c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^{[n]})], \tag{7.16}$$

where $\mathcal{S}_{\gamma/2}$ is the soft-shrinkage operator with $\gamma/2$ as the parameter.

Algorithm 4 presents a pseudocode for the CDL method. In this pseudocode, the outer loop alternates between the optimizing parameters. The inner loops are for updating each parameter, for a given number of iterations, before switching to the other parameter. It is also possible to choose different methods for switching between optimizing parameters. For example one can update with respect to each parameter until the difference between two consecutive updates gets smaller than some small positive value.

7.5 Simulations

In this section, the performance of the proposed CDL method is demonstrated by some experiments. In Chapter 4, it was demonstrated that bounded Frobenius-norm dictionary learning improves the audio coding performance in terms of R-D. A big disadvantage of sparse audio coding using such a learned dictionary is its heavy computational demand, where the real-time implementation on conventional computers is impossible. An alternative to this is to use CDL and a structured $\mathbf{\Psi}$ for a fast implementation.

Algorithm 4 : $\mathcal{CDL}(\mathbf{X}_0, \mathbf{\Psi}_0)$

initialization: $c_\Phi > \|\mathbf{\Phi}^T\mathbf{\Phi}\|$, $K_X, K_\Psi \in \mathbb{N}$
for $t = 0$ **to** T **do**
  $c_\Psi > \|\mathbf{\Psi}^T\mathbf{\Psi}\|$, $\mathbf{X}^{[0]} = \mathbf{X}_t$
  for $n = 0$ **to** $K_X - 1$ **do**
    $\mathbf{X}^{[n+1]} = \mathcal{S}_{\lambda/2}\,[\frac{1}{c_\Phi c_\Psi}(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}\mathbf{X}^{[n]}) + c_\Phi c_\Psi \mathbf{X}^{[n]})]$
  end for
  $\mathbf{X}_{t+1} = \mathbf{X}^{[K_X]}$
  $c_\mathbf{X} > \|\mathbf{X}\mathbf{X}^T\|$, $\mathbf{\Psi}^{[0]} = \mathbf{\Psi}_t$
  for $n = 0$ **to** $K_\Psi - 1$ **do**
    $\mathbf{\Psi}^{[n+1]} = \mathcal{S}_{\gamma/2}\,[\frac{1}{c_\Phi c_\mathbf{X}}(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Psi}^{[n]}\mathbf{X})\mathbf{X}^T + c_\Phi c_\mathbf{X}\mathbf{\Psi}^{[n]})]$
  end for
  $\mathbf{\Psi}_{t+1} = \mathbf{\Psi}^{[K_\Psi]}$
end for
output: $\mathbf{\Psi}_T$

CDL Parameters											
Name	d	M	N	L	λ	γ	$K_\mathbf{X}$	K_Ψ	T	$\mathbf{\Psi}_0$	$\mathbf{X}_0$
Value	256	512	512	8192	0.02	0.01	1	1	1000	$\mathcal{N}(0,1)$	$\mathbf{0}$

Table 7.1: *The CDL parameters in the dictionary learning for sparse audio coding.*

Table 7.1 shows the parameters have been used in this simulation. A two times overcomplete MDCT mother dictionary was chosen as $\mathbf{\Phi}$. The training matrix $\mathbf{Y}$ was generated using random block selection of the same audio signal used in Chapter 5. The parameters c_Φ, c_Ψ and c_X are chosen to be larger but close to the corresponding operator norms, to accelerate the convergence of CDL.

The generation of a selected atom in the learned dictionary $\mathbf{D}$ is schematically demonstrated in Figure 7.1. ψ_i is plotted in part (a). The sparseness of ψ_i is clearly shown. This sparse vector is multiplied with $\mathbf{\Phi}$ to generate one atom of $\mathbf{D}$. Therefore the atoms of $\mathbf{\Phi}$ which are related to the non-zero coefficients of ψ_i contribute to generate $\mathbf{d}_i$. The plots (b), (c) and (d) demonstrate, respectively, the contributing atoms of $\mathbf{\Phi}$, scaled versions of these atoms and $\mathbf{d}_i$.

Now that we have found the learned dictionary, we can show its advantages in the sparse approximation of the audio signals. We chose 4096 different random blocks of samples from the same audio sample. The iterative thresholding method, see Subsection 3.4.2.1, was used for sparse matrix approximation, using $\lambda = 0.02$. An extra step of CDL is re-normalizing the learned dictionary to the initial Frobenius-norm, to make further comparison fair. Figure 7.2 shows the sparsity vs. approximation error plot of the algorithm, in which the horizontal and

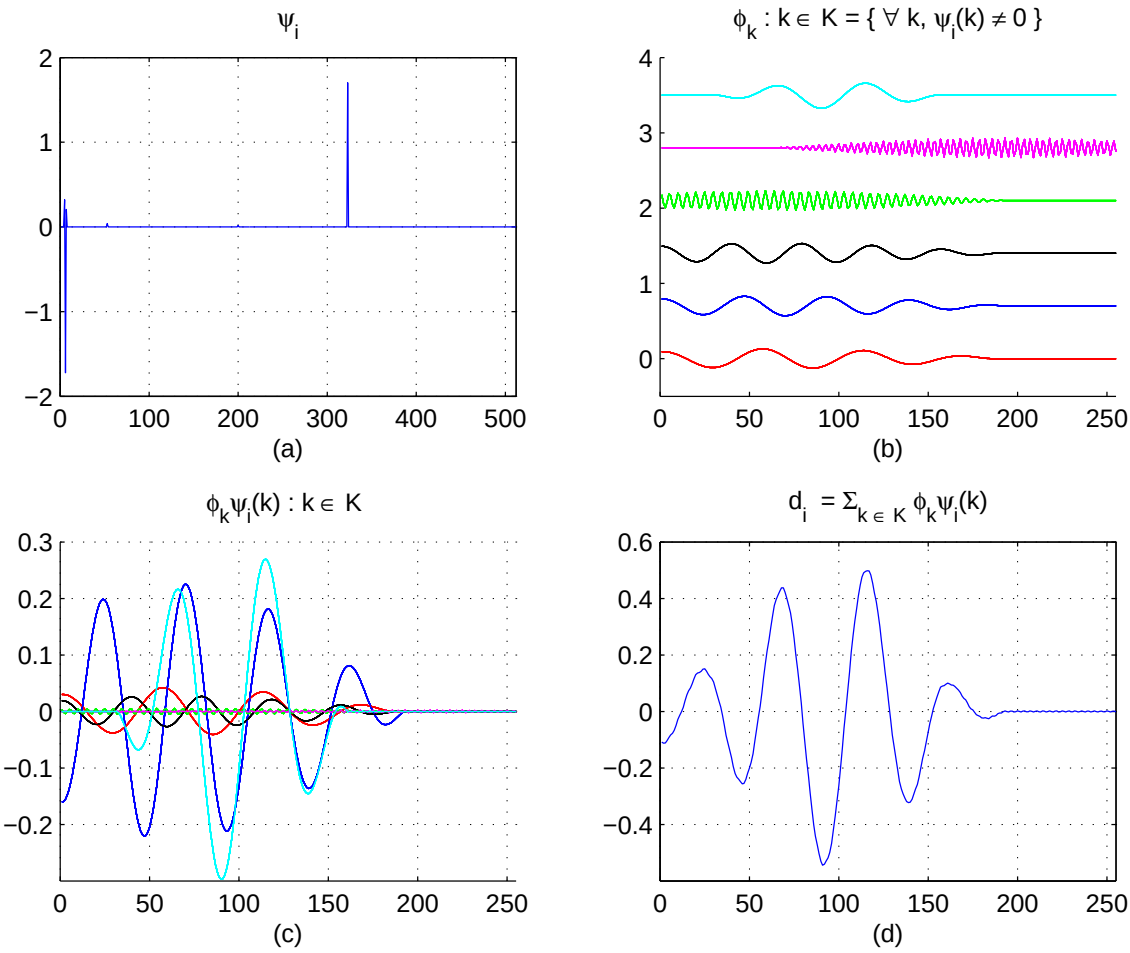

Figure 7.1: *The atom generation in the CDL framework: (a) i^{th} column of $\mathbf{\Psi}$, ψ_i, (b) The atoms ϕ_k which are related to the non-zero values of selected ψ_i, $\{\phi_k : \psi_i(k) \neq 0\}$, (c) $\phi_k\psi_i(k) : \psi_i(k) \neq 0$, (d) The i^{th} atom of $\mathbf{D} = \mathbf{\Phi\Psi}$.*

vertical axes are $\mathcal{J}_1(\mathbf{X})$ and $\|\mathbf{Y} - \mathbf{DX}\|_F^2$, respectively. The result shows that for the similar approximation error, the approximation by using learned dictionary is significantly sparser (has less ℓ_1).

7.6 Summary

In this chapter, a novel dictionary model was introduced. The dictionaries in this model, called compressible dictionaries, are more suitable for fast implementation. Because the approximation complexity of these dictionaries, in comparison to the unstructured dictionaries, is sig-

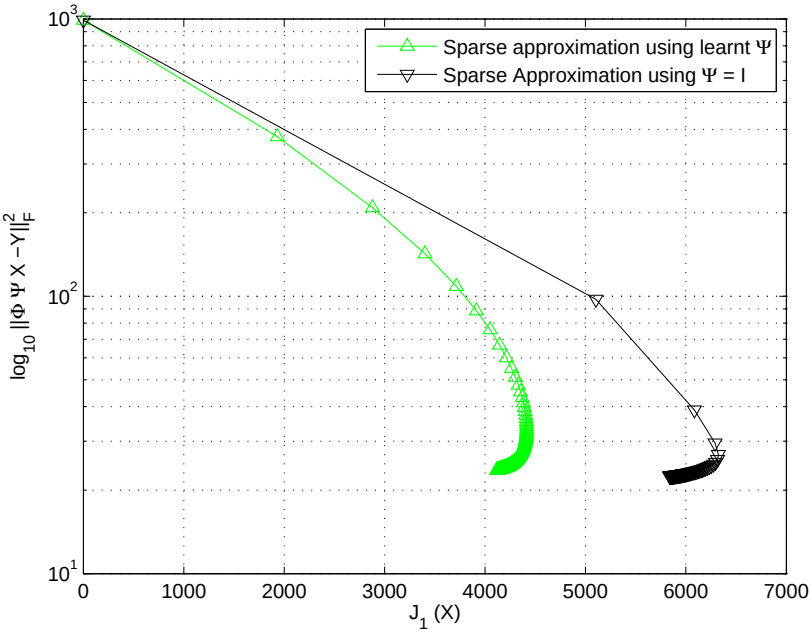

Figure 7.2: *The sparsity $\mathcal{J}_1(.)$ vs. representation error plots of 4096 evaluation signals.*

nificantly less, although there is no analytical studies, fewer training samples appear to be necessary for the dictionary learning. An optimization problem is also formulated in which the optimum dictionary is being close to be compressible. The objective of this optimization problem is non-convex and non-differentiable. A practical algorithm was introduced to find a local minimum. This dictionary learning method, called CDL, can easily be shown, using the Appendix B, to converge to a set of fixed points. As an example a compressible dictionary was learned for the audio signals. It was shown that a sparser approximation of the evaluation samples was obtained by using the CDL learned dictionary. Further investigations on the recoverability and the parameter selection are left for a future work.

Chapter 8
Parametric Dictionary Design (PDD)

8.1 Introduction

The dictionary selection problem is revisited in this chapter to present some motivations for the parametric dictionary design framework. In the previous chapters, the dictionary learning methods were explained where they start with an initial dictionary and refine it by using a set of training samples. In this chapter, a new problem, called parametric dictionary design, is introduced to *design* a dictionary for sparse coding. The domain knowledge is now incorporated into a parametric function, which generates atoms, an admissible set of parameters with an initial set of parameters. The PDD refines the parameters such that the dictionary has a minimal coherence. Minimizing the coherence of the dictionary indirectly helps the practical sparse coding methods to deliver sparser approximation/representation.

8.2 Dictionary Design for Sparse Coding

Let the generative models of the sparse representations and approximations respectively be (2.1) and (2.5). The dictionary $\mathbf{D}$ has been often selected by concatenation of some orthogonal bases [GN03] or using a tight frame [CD05]. These dictionaries can be improved using dictionary learning methods [OF97, LS00, KMR+03, AEB06, YBD09]. The aim of these methods are to adapt the dictionary such that an input signal, taken from a given class of signals, has a sparser approximation. There is an alternative for the dictionary selection, which is called dictionary *design*. Few methods exist to design a suitable $\mathbf{D}$ for a set of natural signals. This could be done by incorporating the knowledge about the *generative model* of the signals into the dictionary design. When the signals are supposed to be understand by the human sensory system, a more efficient method to design $\mathbf{D}$ can be inspired by a human perception model [PAG95], [Dau80]. In this framework, the stimuli responses of the human perception generate some elementary functions, which can be used as the generative model. In fact these elementary functions are more related to the analysis dictionary [EMR07]. These elementary functions

have also been used for generating the synthesis dictionary $\mathbf{D}$. Here, we assume that the set of elementary functions can be described by using a set of parameters and a parametric function. For example, in the multiscale Gabor functions [MZ93], the parameters are scale, time and frequency shifts and the parametric function is Gaussian. In general the parameters are in the continuous domain. To generate a dictionary based upon these generative functions, we can sample these continuous parameters. The question is then how best to sample the parameters. Several researchers have introduced different methods to optimize the sampling process. In [Lee96], a sampling scheme was introduced which finds an approximately tight frame, using 2D Gabor functions.

In a different context, Gammatone and Gammachirp filter banks have been shown to approximate the human auditory system. [KDL07] presented two types of filters which approximate the Gammatone filter banks and allow a possible fast VLSI implementations. Alternatively, some researchers have optimized the parameters based on the closeness to what is observed in the perceptual systems [IP97], [PNHR88], [TGB04]. In practice, [SM08] showed that the optimal parameters, found by fitting to the human auditory system, do not match the parameters are learned from English speech signals in [SL06].

When we use an approximate or a relaxed method, having an exact generative model does not guarantee that we find the best sparse approximation. An important parameter of a dictionary, for a successful sparse recovery, is the coherence μ [Tro04a]. The coherence, which will be explained further in (8.1), is defined as the absolute value of the largest inner-product of two distinct atoms and it has been shown that when μ is smaller than a certain threshold MP and BPDN can recover the sparse representation of the input signal [GMS03], [Tro06b]. It has also been shown that the coherence of a dictionary upper-bounds the residual error decay in MP [GV06] and OMP [Tro04a] and a dictionary with small μ is desirable for sparse coding, where the upper-bound damps faster for such a dictionary. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of the dictionary. The coherence of $\mathbf{D}$ is the maximum absolute value of the off-diagonal elements of $\mathbf{G}$, whenever the columns of the dictionary are normalized. For such $\mathbf{D}$, if the magnitudes of all off-diagonal elements of $\mathbf{G}$ are equal, $\mathbf{D}$ has a minimum coherence [TDHJS05]. This normalized dictionary is called an Equiangular Tight Frame (ETF) [STDH07]. Although this type of frame has various nice properties, we mainly consider the advantages in exact atom recovery [Tro04a] and the residual error decay rate [GV06]. Unfortunately ETF's do not exist for any arbitrary selection of d and N [STDH07]. Therefore a dictionary design aim can

be to find the nearest admissible solution. On the other hand, natural signals do not generally have sparse approximations using an ETF dictionary. The dictionary design problem can now be defined as "finding a parametric dictionary whose Gram matrix is close to being the Gram matrix of an ETF". This way, domain knowledge is incorporated into the parametric functions used, while the optimization aims at improving the ability of algorithms to find sparse approximations. We expect that the given class of signals has a sparse approximation using such a dictionary, as it is generated by sampling the parameters of the given generative function, whilst the dictionary has an extra property to be close to being an ETF. In practice it has been shown that the designed dictionary indeed gives advantages over the standard dictionary, in terms of efficient sparse approximation. Another advantage of the parametric dictionary is that sparse approximation methods only need to store the parameters, instead of the full dictionary, which offers a huge reduction in memory requirement (the size of parameter matrix is much smaller than the size of the corresponding dictionary). Sometimes this type of parametric dictionary can furthermore be multiplied to the coefficient vectors faster than direct matrix-vector multiplication. It then also speeds up most of the currently available sparse coding methods.

The proposed parametric dictionary design has also some dis-advantages. PDD is not a sample based dictionary selection method. For example if the actual data samples are accumulated in a subspace of the signal space, the dictionary design method will not be able to exploit this feature since it attempts to find a dictionary that spans the whole space uniformly. We also assume in the parametric dictionary design that our signals will be equally well modelled in all forms of the parametric dictionary, which is not very accurate assumption.

A difficulty in the given practical method is that the current algorithm stores the Gram matrix explicitly. Therefore the current method is not tractable for a very large dictionaries.

In the next section, the parametric dictionary design problem is formulated, followed by presenting a practical algorithm to find an approximate solution. The PDD is applied to a case study in which the gradient formula derivation is done in Appendix C. Experiments, in the simulation subsection, show the advantages of the proposed dictionary design. The stability and convergence analysis of the algorithm is presented in Appendix D.

8.3 PDD: Formulation

The problem of finding $\mathbf{D}$ that is close to being an ETF is formulated in this section. Let $\mathbf{D}_\Gamma \in \mathscr{D}$ be a parametric dictionary. Γ is the parameter matrix, with γ_i as its i^{th} column and $\mathscr{D}$ is the set of admissible parametric dictionaries. By letting $\mathbf{D}_\Gamma$ be a matrix with atoms $\mathbf{d}_i$ (with the associated parameters γ_i), we implicitly assume that the generative model is discrete. This model can be extended to a continuous model, which is out of scope of this thesis. To select a $\Gamma \in \Upsilon$, where Υ is an admissible parameter set, we can optimize an objective function. In section 8.1 we explained that for a better performance in sparse coding, we are interested to design a dictionary which is close to being an ETF. For a given normalized $\mathbf{D}$, the coherence of $\mathbf{D}$, $\mu_\mathbf{D}$, is defined by,

$$\mu_\mathbf{D} = \max_{i,j:j\neq i} \{|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|\}. \tag{8.1}$$

A column normalized dictionary $\mathbf{D}_G$ is called ETF, or Grassmannian frame [SH03], when there is a $\gamma \ : \ 0 < \gamma < \pi/2$.

$$|\langle \mathbf{d}_i, \mathbf{d}_j \rangle| = \cos(\gamma) \ : \ \forall i, j \ i \neq j \tag{8.2}$$

Strohmer et. al. in [SH03] showed that if there exists an ETF in $\mathscr{D}$, here the set of d by N uniform frames[1], it is the solution of,

$$\arg\min_{\mathbf{D}\in\mathscr{D}} \{\mu_\mathbf{D}\}. \tag{8.3}$$

To study the lower bound of μ_D, the existence of an ETF and its Gram matrix, [SH03] introduced the following Theorem.

***Theorem* 8.3.1.** *[SH03, Theorem 2.3] Let* $\mathbf{D}$ *be a uniform frame in* $\mathbb{R}^{d\times N}$*. Then*

$$\mu_\mathbf{D} \geq \mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{8.4}$$

Equality holds in (8.4) if and only if $\mathbf{D}$ *is an ETF. Furthermore, equality in (8.4) can only hold if* $N \leq \frac{d(d+1)}{2}$.

Let Θ_d^N be the set of Gram matrices of all $d \times N$ ETF's. If $\mathbf{G}_G \in \Theta_d^N$ then the diagonal elements and the absolute values of the off-diagonal elements of $\mathbf{G}_G$ are one and μ_G respectively. A nearness measure of $\mathbf{D} \in \mathbb{R}^{d\times N}$ to the set of ETF's can be defined as the minimum distance

[1] A frame with unit column norms.

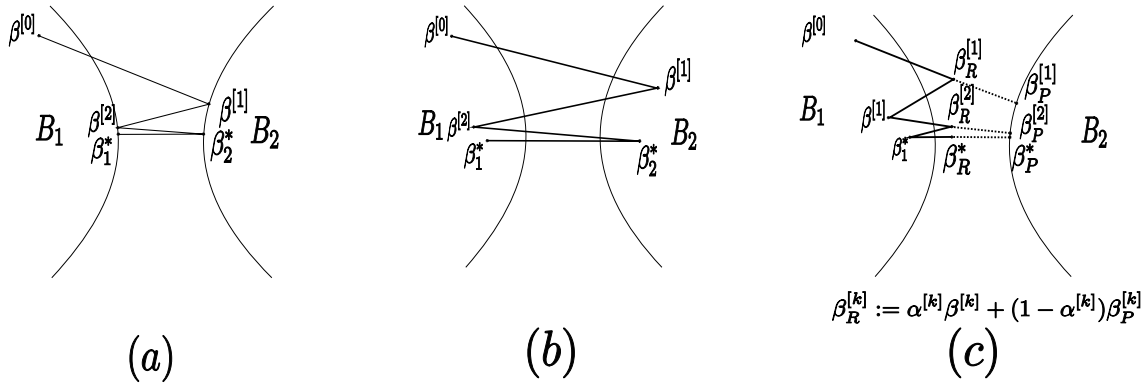

Figure 8.1: *Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method.*

between the Gram matrix of $\mathbf{D}$ and $\mathbf{G}_G \in \Theta_d^N$ [TDHJS05]. To optimize the distance of a dictionary to an ETF, we can solve,

$$\inf_{\Gamma \in \Upsilon, \mathbf{G}_G \in \Theta_d^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_G\|_\infty, \tag{8.5}$$

where the matrix operator $\|.\|_\infty$ is defined as the maximum absolute value of the elements of the matrix. Instead, we would like to use a different norm space which simplifies the problem[2]. An advantage of using ℓ_2 measure in the given problem is that it considers the errors of all elements (and not just the maximum absolute error). In this setting, when there is no ETF in $\mathscr{D}$, we find a dictionary that is close to be quasi-incoherent [Tro04a] [GV06]. Therefore we use the following formulation,

$$\inf_{\Gamma \in \Upsilon, \mathbf{G}_G \in \Theta_d^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_G\|_F^2. \tag{8.6}$$

This is generally a non-convex optimization problem, which might have a set of solutions or no solution (e.g. Θ_d^N is empty as there do not always exist ETF's for arbitrary N and d). To avoid this complication, one can extend Θ_d^N to a convex set Λ^N [TDHJS05], which is non-empty for any N, by

$$\Lambda^N = \{\mathbf{G} \in \mathbb{R}^{N \times N} : \mathbf{G} = \mathbf{G}^T, \operatorname{diag} \mathbf{G} = 1, \max_{i \neq j} |g_{i,j}| \leq \mu_G\}. \tag{8.7}$$

Note that now there exists $\mathbf{G} \in \Lambda^N$ which is not a Gram matrix for $\mathbf{D} \in \mathbb{R}^{d \times N}$. Relaxing (8.6),

[2]Although the matrix space with ℓ_∞ is a well defined Banach space, we here use ℓ_2 norm Hilbert space to use easy formulation of the optimization process.

Algorithm 5 *Parametric Dictionary Design*

initialization: $k = 1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1 \leq i \leq K} : 0 < \alpha_i \leq 1$
while $k \leq K$ **do**
  $\mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
  $\mathbf{G}_{P_{k+1}} = \min_{\mathbf{G} \in \Lambda^N} \|\mathbf{G}_{\Gamma_k} - \mathbf{G}\|_F$
  $\mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1 - \alpha_k)\mathbf{G}_{\Gamma_k}$
  $\mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : \|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}\|_F < \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\}$
  $k = k + 1$
end while

by replacing Θ_d^N with Λ^N, gives the following optimization problem.

$$\inf_{\Gamma \in \Upsilon, \mathbf{G} \in \Lambda^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{8.8}$$

An important difference between (8.6) and (8.8) is that the relaxed problem is guaranteed to have at least a solution. We therefore use the relaxed formulation from now on. We show experimentally that the approximate solutions of (8.8), even though the Gram matrix of the dictionary might only be close to Λ^N, show good performances in sparse approximation.

In the next section a practical method is presented to find an approximate solution for (8.8). Our approach has similarities with alternating minimization. This method is guaranteed not to increase the objective function in each step. Because the objective is non-negative, the algorithm is stable due to Lyapunov's second theorem [Lya66] and one can also show that the objective function converges. Therefore, the stability of the algorithm and the convergence of the objective function do not prove the convergence of the algorithm. Appendix D shows that, under certain conditions, the algorithm converges to a set of fixed points.

8.4 PDD: A Practical Algorithm

A standard method to solve (8.8) is alternating projection, see for example [SY98], [TDHJS05] and references therein. In this method we alternatingly project the current solution onto the admissible sets, see Fig. 8.1.a. When the admissible sets are convex, the algorithm converges[3] to a solution in $\mathscr{D} \cap \Lambda^N$ or, when $\mathscr{D} \cap \Lambda^N = \emptyset$, to a pair of solutions respectively in $\mathscr{D}$ and Λ^N. In the following, a formulation for the projection onto Λ^N is derived, but there is no easy formulation for the projection onto the set of admissible dictionaries, in general. Therefore a

[3] At least in finite dimensional spaces. There are counter-examples for the lack of convergences in the infinite dimension setting [HH04].

different method has been chosen which has similarities with alternating minimization [CT84] (or generalized alternating projection [GB05]), see Fig. 8.1.b. In the alternating minimization framework, we choose the new solutions in $\mathscr{D}$ and Λ^N alternatingly such that the objective does not increase in each update and is thus stable.

Although the proposed algorithm has similarities with alternating minimization, it does not follow its steps exactly. The difference is that in the stage in which we update the current solution with respect to Λ^N, we choose a point which is somewhere between the current solution and the projection onto Λ^N. Fig. 8.1.c shows a schematic representation of the proposed method. The reason for this modification is that by projection onto Λ^N, the structure of the Gram matrix changes significantly so that the selection of a new point in $\mathscr{D}$ in the following step is very effective. We can gradually select a closer point to the projected point on Λ^N, when the current $\mathbf{D}_\Gamma$ is close to Λ^N. In the other step, we update $\mathbf{D}$ such that it does not increase the objective in (8.8).

The parametric dictionary design is summarized in Algorithm 5. In line 5, the algorithm finds the projection onto Λ^N. In line 5, a point in $\mathscr{D}$ is selected which is closer to $\mathbf{G}_{R_{k+1}}$. In the following we show how we calculate the updates in lines 5 and 5.

8.4.1 Projection onto Λ^N:

In the objective function of (8.8), $\mathbf{G}$ is a Hermitian matrix. By sign change of any related off-diagonal pair of elements , i.e. $g_{i,j}$ and $g_{j,i}$, we get a new $\widetilde{\mathbf{G}} \in \Lambda^N$. The closest $\mathbf{G}$ to $\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma$, in a Frobenius norm space, is the $\mathbf{G}$ with a similar sign pattern. We know that in a normed space, finding the nearest element of a set to a point is a projection of that point onto the set. Because Λ^N is convex, the projection is unique. For a given $\mathbf{G}_\mathbf{D} = \mathbf{D}^T\mathbf{D} : \mathbf{D} \in \mathbb{R}^{d\times N}$, the projection of $\mathbf{G}_D$ onto Λ^N can be found by the following operator [TDHJS05],

$$g_{P\,i,j} = \begin{cases} \text{sign}(g_{D\,i,j})\mu_G & i \neq j \\ 1 & otherwise\ , \end{cases} \tag{8.9}$$

where μ_G is as defined in (8.4). This operator can be used to find $\mathbf{G}_{P_{k+1}}$ in line 5 of Algorithm 5, by applying it to $\mathbf{G}_{\Gamma_k}$.

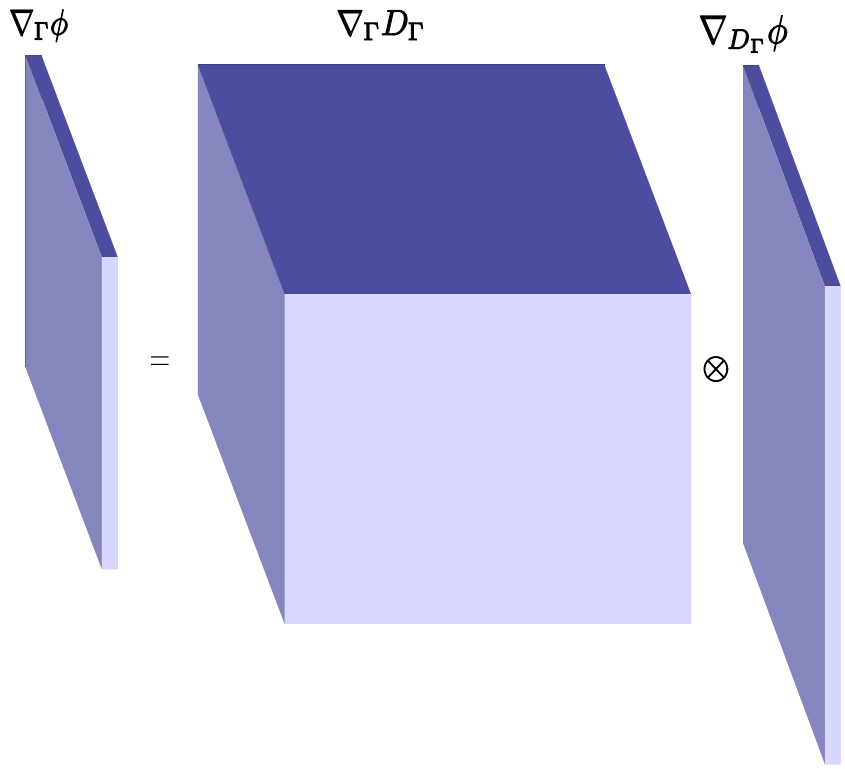

Figure 8.2: *The chain rule (8.11) in the tensor form.*

8.4.2 Parameter update:

Let us assume $\mathbf{D}_\Gamma$ is a differentiable function on Υ and therefore (8.8) is a differentiable function on Υ. An easy way to find Γ_{k+1}, such that it satisfies line 5 of the Algorithm 5, is to use the gradient descent method. We rewrite (8.8) as a minimization problem based on Γ when $\mathbf{G}_{R_{k+1}}$ is fixed.

$$\min_{\Gamma \in \Upsilon} \phi(\Gamma) \ , \ \phi(\Gamma) := \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}\|_F^2 \tag{8.10}$$

The gradient of the objective function in (8.10) can be found by chain rule for the matrix functions [Dat09, D.1.3].

$$\begin{aligned} \nabla_\Gamma \phi &= \nabla_\Gamma \mathbf{D}_\Gamma \ \nabla_{\mathbf{D}_\Gamma} \phi \\ &= \nabla_\Gamma \mathbf{D}_\Gamma \ \nabla_{\mathbf{D}_\Gamma} \operatorname{tr}\{\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma \mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - 2\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma \mathbf{G}_{R_{k+1}} + \mathbf{G}_{R_{k+1}} \mathbf{G}_{R_{k+1}}\} \\ &= 4(\nabla_\Gamma \mathbf{D}_\Gamma)\, \mathbf{D}_\Gamma (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}) \end{aligned} \tag{8.11}$$

In this formulation, one still needs to calculate $\nabla_\Gamma \mathbf{D}_\Gamma$. In the appendix C, we derive this formulation for a special parametric dictionary. We iteratively use the gradient descent method to find a *local* minimum of the problem (8.10). Let $\Gamma_k^{[0]} = \Gamma_k$, the updating formula is as

follows,

$$\Gamma_{k+1}^{[l+1]} = \Gamma_{k}^{[l]} - \epsilon \nabla_{\Gamma} \phi \big|_{\Gamma_{k}^{[l]}}, \tag{8.12}$$

where ϵ is a small positive value. The parameter ϵ should be chosen such that the update reduces the objective function in (8.10) [Fle87]. In this framework, $\Gamma_{k+1} = \lim_{l \to \infty} \Gamma_{k+1}^{[l]}$. In practice we stop after a given number of iterations or when $\epsilon \nabla_{\Gamma} \phi \big|_{\Gamma_{k}^{[l]}}$ becomes very small. Algorithm 6 summarizes this parameter update algorithm.

Because $\phi(\Gamma)$ is continuous, its epigraph [BV04], for an initial Γ_0[4], is closed. By choosing a bounded set of admissible parameters Υ, the epigraph is a compact set in Euclidean space. To show that the algorithm gets as close as possible to the set of limit points, we need to use the Bolzano-Weierstrass theorem.

***Theorem* 8.4.1.** *[Apo74, 3.24] Every bounded infinite subset of $\mathbb{R}^N$ has at least one limit point in $\mathbb{R}^N$.*

Therefore, when the set of admissible parameters is bounded and ϵ is selected such that moving in the gradient direction with this step size reduces the objective, this gradient descent algorithm has at least one limit point in the admissible set.

***Remark* 8.4.1.** The function $\phi(\Gamma)$ is a lower bounded function. Hence, if we reduce ϕ in each iteration, due to the Lyapunov's second theorem [Lya66], the algorithm is stable.

***Remark* 8.4.2.** Algorithm 5 is an iterative algorithm in which we also used another iterative method for the dictionary update in line 5. The stability and the convergence of the updates mentioned above were related to the inner loop in Algorithm 5. The convergence of Algorithm 5 is studied in Appendix D.

***Remark* 8.4.3.** We draw the readers attention to the formulation (8.11). The parameters $\nabla_{\Gamma} \mathbf{D}_{\Gamma}$, $\nabla_{\mathbf{D}_{\Gamma}} \phi$ and $\nabla_{\Gamma} \phi$ are tensors of rank 3, 2 and 2 respectively. If $\Gamma \in \mathbb{R}^{p \times N}$ and $\mathbf{D} \in \mathbb{R}^{d \times N}$ then $\nabla_{\Gamma} \mathbf{D}_{\Gamma} \in \mathbb{R}^{p \times d \times N}$, $\nabla_{\mathbf{D}_{\Gamma}} \phi \in \mathbb{R}^{d \times 1 \times N}$ and $\nabla_{\Gamma} \phi \in \mathbb{R}^{p \times 1 \times N}$. A graphical presentation of this formulation is presented in Fig. 8.2 . Furthermore, to use this directional update in (8.12), we need to map $\nabla_{\Gamma} \phi \in \mathbb{R}^{p \times 1 \times N}$ into the appropriate matrix in $\mathbb{R}^{p \times N}$. It is easily done by changing the order of indices (1,2,3 to 1,3,2), following by cancelling the third dimension. Because the rank of $\nabla_{\Gamma} \phi$ is 2, this mapping is injective.

[4]Epigraph of $\phi(\Gamma) : \Upsilon \to \mathbb{R}$ for an initial Γ_0 is defined [BV04, 3.1.7] by: $\operatorname{epi} \phi = \{\Gamma \, : \, \Gamma \in \Upsilon, \phi(\Gamma) \leq \phi(\Gamma_0)\}$

Algorithm 6 $Parameters\ Update$

1: **initialization:** $l = 1$, $1 \leq L$, $\Gamma_k^{[0]} = \Gamma_k$, $\epsilon \in \mathbb{R}^+$, $\phi(\Gamma) = \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}\|_F^2$
2: **for all** $l \leq L$ **do**
3: $\quad \Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$
4: $\quad l = l + 1$
5: **end for**
6: $\Gamma_{k+1} = \Gamma_{k+1}^{[L]}$

8.5 Case study

The formulated problem in Section 8.3 has been developed in a general form. To show the advantages of using parametric dictionary design, a case study is chosen. In sparse audio processing, an important question is how to choose the dictionary [DD06], [RRD08b]. Different methods have been introduced to adapt the dictionary to better fit a set of training samples [SL05], [Lew02], [YBD09]. Alternatively, some researchers used a class of parametric dictionaries based on Gammatone filter banks, which have been shown to have similarities with the human auditory system [SM08], [PNT07]. The following subsections are proposed to show that the parametric dictionary design improves the performance of audio sparse approximation and exact recovery based around a Gammatone representation.

8.5.1 Gammatone parametric dictionary

The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1} e^{-2\pi bBt} \cos(2\pi f_c t) \tag{8.13}$$

where $B = f_c/Q + b_{min}$, f_c is the centre frequency and $n \in \mathbb{N}$, a, b, Q, b_{min} are some constants. The optimal parameter selection is not easy. One can select the parameters such that the generated atoms match the auditory impulse response. The auditory system has been optimized through evolution for an optimal perception. This system may not provide the optimum model for a specific application like sparse generative model. Our goal is to optimally select these parameters so that sparse approximation methods can be used more efficiently. Another difficulty in using the Gammatone filter banks as a dictionary is its large size. A moderate size dictionary can be designed by the proposed method.

The dictionary is generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift.

Here, $\gamma = [t_c \; f_c \; n \; b]^T$ are the optimization parameters. The parameters t_c and f_c change the center of the atoms in the time-frequency plane. n and b control the rise time and the width of the atoms in the time domain, respectively. The parameter a is chosen to normalize the norm of each atom. Let $\{\gamma_i\}_{1\leq i\leq N}$ be a set of the parameters and $g_{\gamma_i}(t)$ be the atom generated using γ_i. The parameter matrix Γ and the parametric dictionary $\mathbf{D}_\Gamma$ are generated using γ_i and $g_{\gamma_i}(\lfloor t f_{samp} \rfloor)$ as the columns respectively, where f_{samp} is the sampling frequency.

To use the method introduced in 8.4.2, $\mathbf{D}_\Gamma$ should be differentiable. $\mathbf{D}_\Gamma$ can be made differentiable with respect to Γ by extending the domain of (8.13) to a more general set using $n \in \mathbb{R}$. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, Υ is become a compact set which guarantees convergence of the algorithm to a set of fixed points (due to Theorem D.0.1). A back-projection into Υ is necessary in Algorithm 5, when at least one parameter goes out of Υ. One should compare the current and the previous solution to make sure that the update step reduce the objective. A simple back-projection operator is thresholding operator, where it chooses the closest admissible parameter.

Although the computation of the gradient of parametric dictionary generated by using $g(t)$ is straight forward, it is derived in the Appendix C for completeness.

8.5.2 Simulations results

The proposed dictionary design method using the Gammatone dictionary discussed in 8.5 is studied in this section. We first investigate the characteristics of the dictionaries throughout the design iterations. The stability of the algorithm is demonstrated by showing that the objective function reduces in each step. In the second part of this subsection, we compare the performance of the initial and the optimized dictionaries in terms of sparse approximation and exact sparse recovery. Gammatone type dictionaries are proposed for sparse approximation of audio and they have been chosen as the examples accordingly. In all the simulations we choose two times overcomplete dictionaries with a window size 1024. (N.B. for this redundancy the existence of an ETF is feasible.)

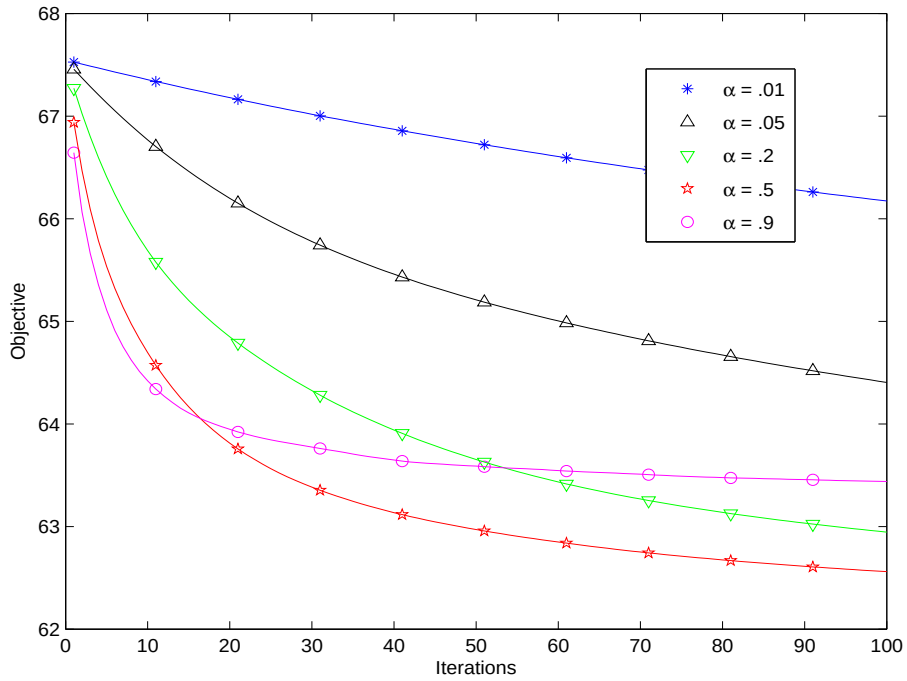

Figure 8.3: *The objective functions for different* $\{\alpha_k\}_{\forall k,\alpha_k=\alpha}$*, for a constant* α*.*

8.5.2.1 Algorithm Evaluation

The given algorithm is evaluated in three different ways. First, it is shown that the algorithm reduces the objective (8.8) in each iteration using different $\{\alpha_k\}_{\forall k,\alpha_k=\alpha}$. The parameter B, defined after (8.13), is the bandwidth of the audio filterbank at the center frequency f_c. The values $n = 4$, $Q = 9.26449$, $b_{min} = 24.7$, as they have been suggested in [GM90] and [Sla88], and $b = 0.65$ are chosen for the simulations. To generate the initial dictionary, f_c and t_c are sampled. The method introduced in [Sla93], to make the filter bank, is used here. In this method an extra parameter δ, called step factor, is introduced to indicate the amount of frequency overlap. In this framework the k^{th} frequency center is calculate using the following formula.

$$f_c^k = -Qb_{min} + (f_s/2 + Qb_{min})e^{-k\delta/Q} \tag{8.14}$$

f_s is the maximum allowed frequency, which is half of the Nyquist frequency. In the simulations, δ is set to be 0.45. A similar method is used to sample t_c. This time sampling is linear, in contrast with the logarithmic sampling in (8.14). Let the peak of the envelope of the impulse response of the filter be at t_p and σ indicate the amount of time overlap. The l^{th} time center is

found using,

$$t_c^l = t_p + \sigma(l-1)\, t_p \,. \tag{8.15}$$

σ is set to be 0.75 in the simulations. It is necessary to mention that t_c^l is implicitly a function of f_c^k. A set of $\{f_c^k\}_{k\in\mathscr{K}}$ is generated at first. For each generated atom using f_c^k and $t_c = 0$, a set of time-shifted versions is then generated using $\{t_c^l\}_{l\in\mathscr{L}}$.

To generate a dictionary, $g_{\gamma_i}(t)$ is windowed to a size equal to the signal length d and it is periodized such that one period is selected as an atom using the following formula,

$$\mathbf{d}_{\gamma_i,j} = \begin{cases} g_{\gamma_i(j+d)} & 1 \le j < j_{c_i} \\ g_{\gamma_i(j)} & j_{c_i} \le j \le d, \end{cases} \tag{8.16}$$

where $j_{c_i} = \lfloor t_{c_i} f_{samp} \rfloor$. A simple sequence of $\{\alpha_k\}$ is selected using $\alpha_k = \alpha$ for all k and a constant α in all simulations. A more complicated sequence might improve the performance of Algorithm 5. However it has not presented here. Instead, it is intended to show that the designed dictionary is superior to the initial dictionary in practice, even with a simple $\{\alpha_k\}$. The effect of α is investigated in the first experiment. The objective function (8.8) for a selected α's is plotted in Fig. 8.3. As we expect, simulations show reduction of the proposed objectives in each iteration. It is also demonstrated that if α is small, the algorithm converges very slowly. Although using a large α is desirable for a fast convergence, the solution is not as good as the solution found by using a medium range α. For other simulations $\alpha = 0.5$ is selected to find a good solution after an acceptable number of iterations.

The proposed algorithm searches for an equiangular *tight frame*. Therefore one way to show the performance of the proposed algorithm is to compare the singular values (SV) of the designed dictionary and a tight frame. A tight frame in $\mathbb{R}^{d\times N}$ has d non-zero SV equal to $\sqrt{N/d}$. We have plotted the sorted SV's of the dictionaries at selected iterations in Fig. 8.4. It can be seen that the SV's of the designed dictionary become closer to the SV's of the tight frame after each selected number of iterations.

Given that the algorithm is based on distances in the Gram matrix domain, another way to evaluate the algorithm is to show the Gram matrix of the dictionary. The ℓ_2 norm of each row of the Gram matrix is plotted in Fig. 8.5. The Gram matrix of the original dictionary and the designed dictionary, after 100 iterations, are respectively shown in the left and right windows. The ℓ_2 norm of a possible ETF is also shown with a dashed line as reference. It can be seen

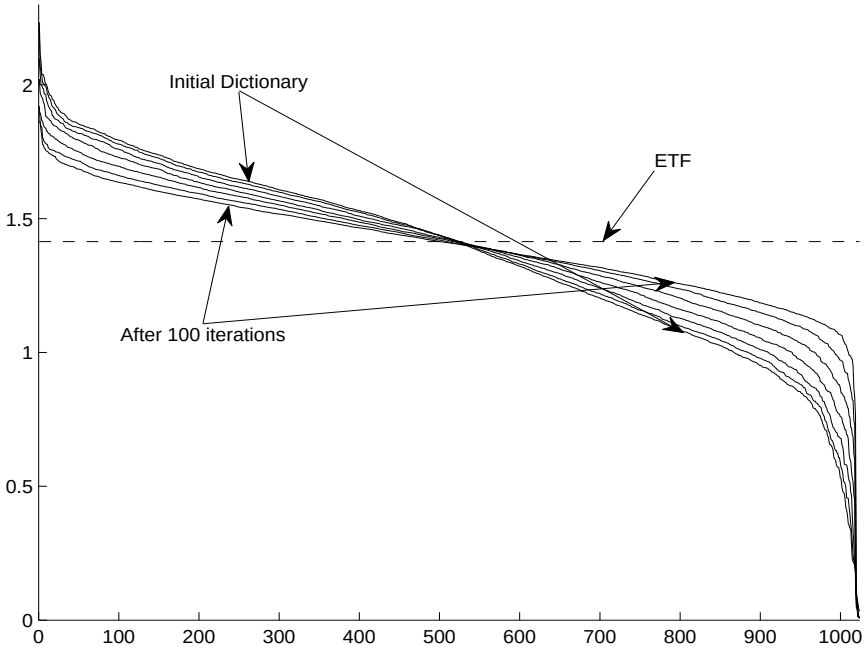

Figure 8.4: *Eigen values plot of the dictionary.*

that the Gram matrix of the designed dictionary is closer to the desired Gram matrix.

This parametric dictionary is attempting to tile the time-frequency plane. An ETF is a frame having the minimum total overlap between atoms, which can here be measured by the magnitude of inner-product of two atoms, but not always having localized representation in the time-frequency plane. A dictionary which is simultaneously an ETF, or closed to being an ETF, and localized in time and frequency, tiles time-frequency plane more uniformly. The Wigner-Ville (WV) time-frequency representation of the atoms are chosen to demonstrate tiling pattern. The contour plots of the original and the designed atoms are respectively shown in Fig. 8.8 and 8.9 using a similar method to that used in [AP01]. Although the algorithm attempts to minimize μ by changing the structure of the dictionary, the locations which are not covered by the high energy part of any atom demonstrate the local minimum convergence of the algorithm. It also shows a potential for a more efficient update operator than the gradient descent Algorithm 6.

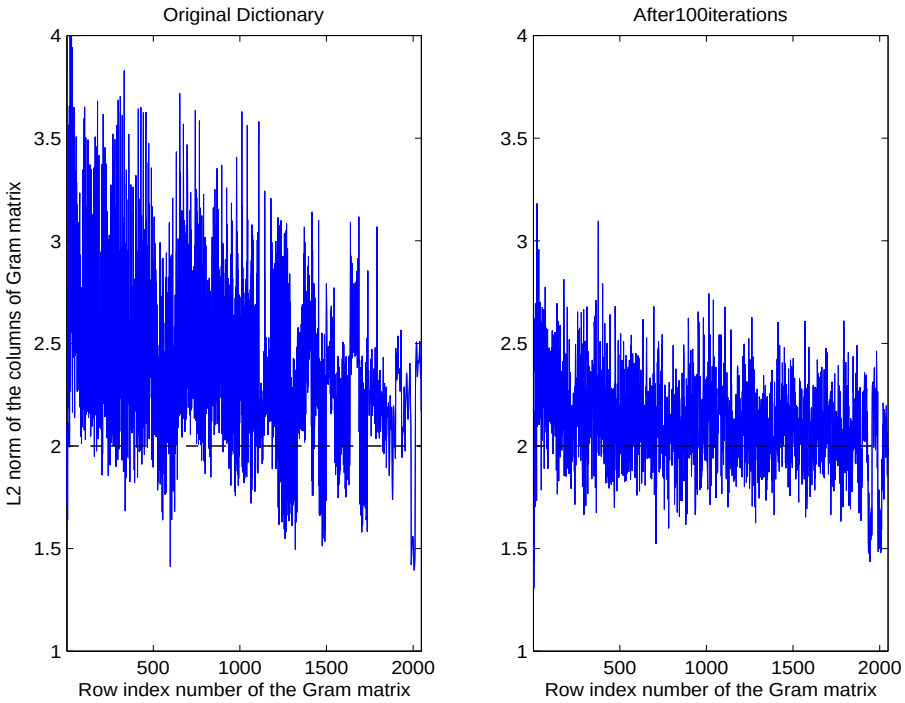

Figure 8.5: *The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries.*

8.5.2.2 Exact sparse recovery and sparse approximation

In this part, the advantages of the parametric dictionary design are demonstrated in terms of exact sparse recovery [Tro04a] and sparse approximation. In the first experiment sparse coefficient vectors, with different sparsity, are generated and the percentages of the exact recovery are plotted for those sparse vectors.

The location of the non-zero coefficients are selected uniformly at random and the PDF of the magnitudes are selected to be Gaussian with zero mean. The Matching Pursuit (MP) algorithm is used to find the sparse approximation. The rate of exact support recovery is calculated by the ratio of the number of correctly found non-zero coefficient places to the number of cases in which at least one location of the zero coefficient is set to be non-zero. The simulations are run for 1000 times. This ratio is shown as the percentage of exact recovery in Fig. 8.6. It is clear that the design method has improved the exact recovery ratio.

For sparse approximation applications, it is more interesting to have a dictionary that, if it fails to satisfy exact recovery [Tro04a], still gives a sparse approximation for a given class of signals.

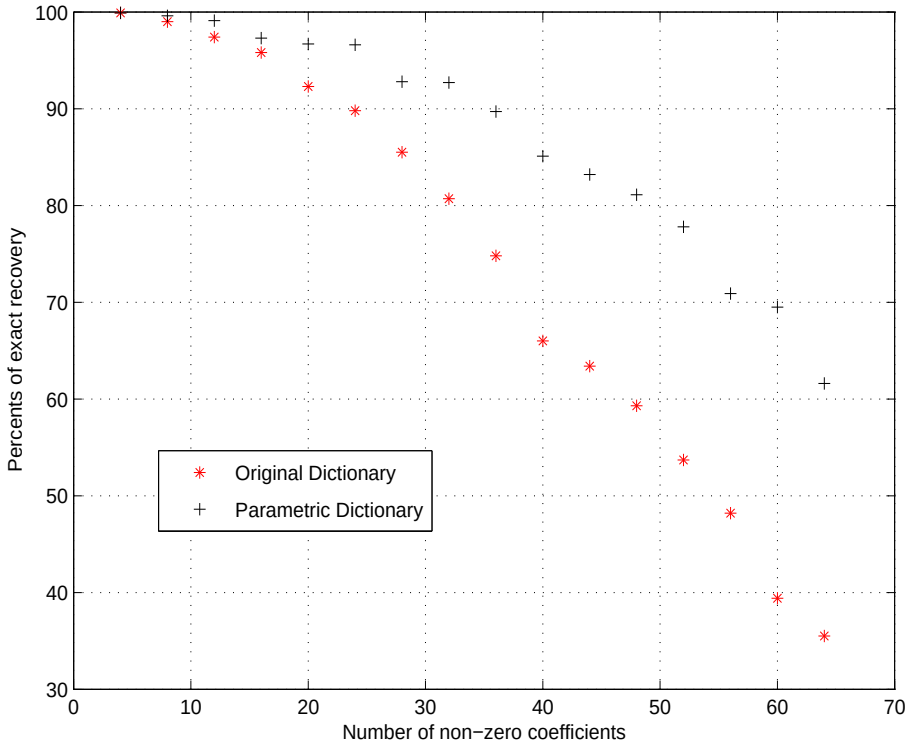

Figure 8.6: *Exact support recovery of the sparse signals.*

Therefore as the second experiment, the decay rate of the residual error is compared when MP is used for sparse approximation [GV06]. An audio signal is taken from more than 8 hours recorded from BBC Radio 3, which mostly plays classic music. It is first down-sampled by a factor of 4 and summed across the stereo channels to make a mono signal with 12K samples per second. The original Gammatone and the parametric designed dictionaries for 100 blocks, each with the length of 1024 samples, are used in this experiment. The average decay rate of the residual errors, in logarithmic scales, are shown in Fig. 8.7. This rate directly influences the performance of the sparse approximation methods. That is, we can better approximate the signal with fewer coefficients using a high residual error decay rate dictionary. In Fig. 8.7, although the curves start with the same slope, after a few iterations, here 10, the designed dictionary shows a clear advantage.

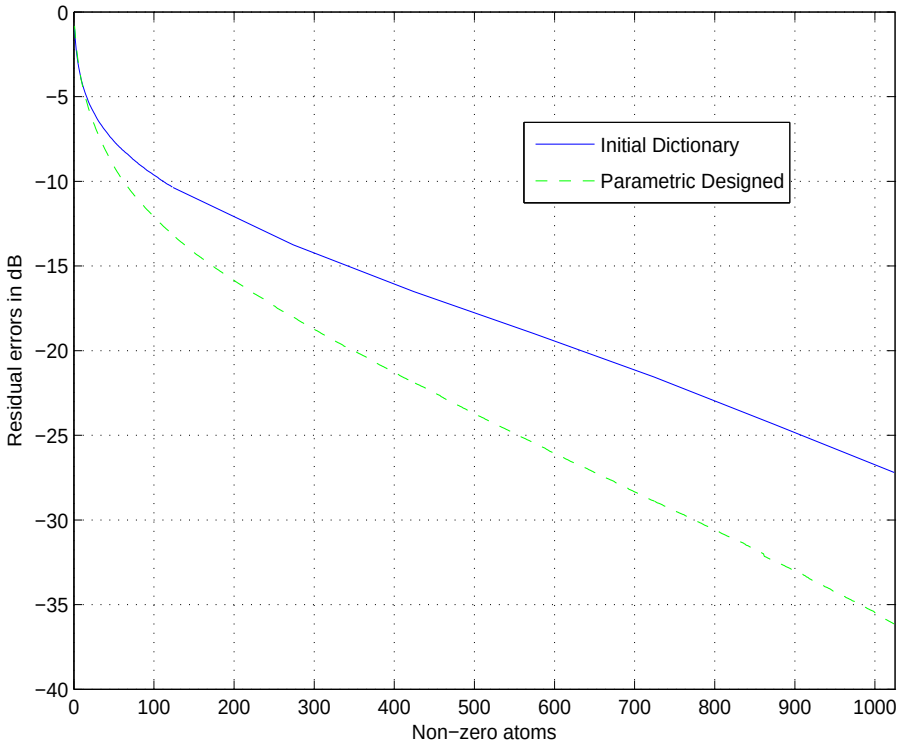

Figure 8.7: *The residual error using matching pursuit for sparse approximation of the audio signal.*

8.6 Summary

The sparse approximation methods successfully approximate a class of signals with a set of sparse coefficient vectors, when an appropriate generative model is given. A signal independent dictionary design method has been formulated in this chapter. In this method a criterion based on an important feature for the success of sparse approximation methods is considered. A priori knowledge about the signal was included by using parametric functions. In this framework it has been shown that the dictionary design problem is to find an optimal set of parameters. This problem can in general not be solved exactly. Fortunately an approximate solution is found using the proposed method. It was shown, by some simulations, that A) the given method can find an appropriate set of parameters for the given case study and B) the designed dictionary showed promising performance advantages in terms of exact recovery and sparse approximation of audio signals. What have been shown in this chapter can be a basis for designing parametric dictionaries under extra constraints, such as to be shift-invariance, quasi-incoherence, data de-

pendence, to have tree structures or structures for fast implementation. However, this has been left for future work.

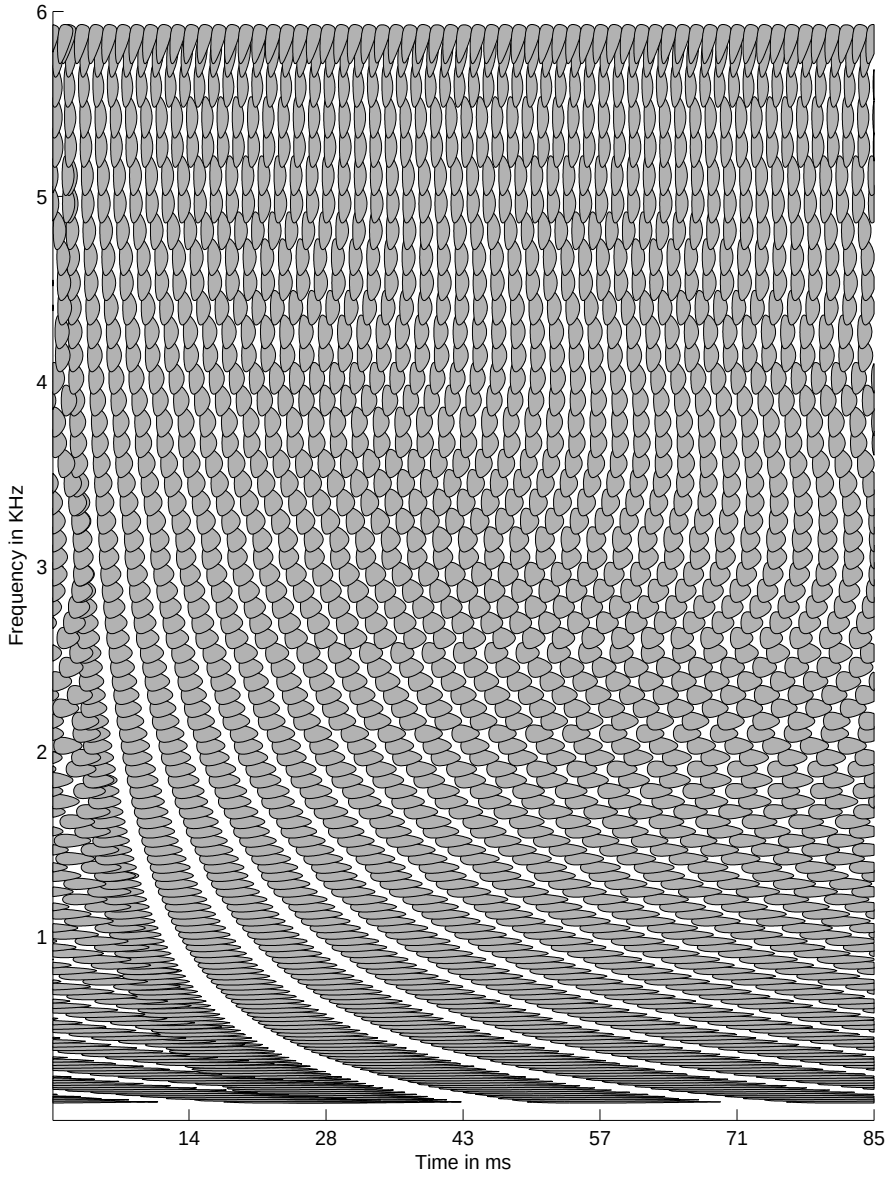

Figure 8.8: *Wigner-Ville contour plots of the original Gammatone atoms. The WV contour of each atom is calculated at 0.7 times it peak.*

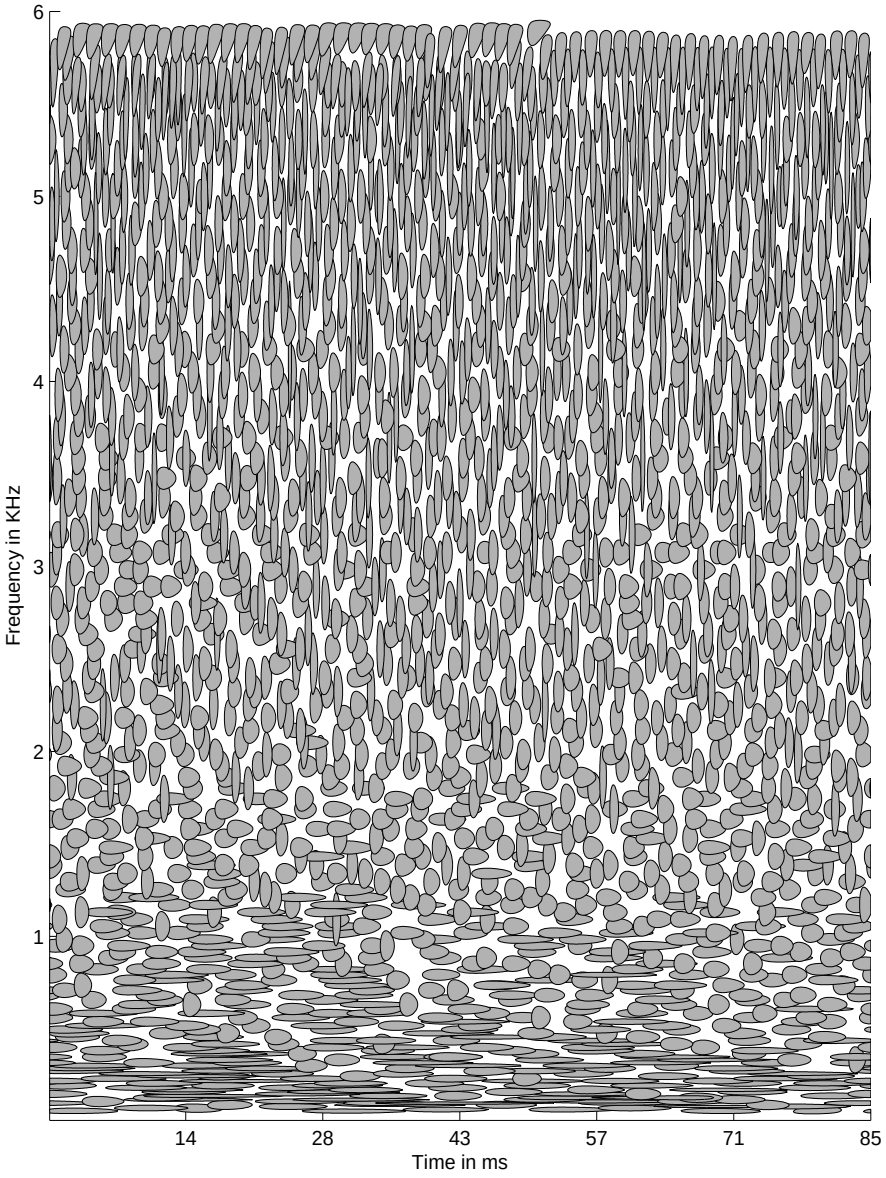

Figure 8.9: *Wigner-Ville contour plots of the learned Gammatone atoms. The contours are calculated similar to Fig. 8.8*

Chapter 9
Conclusion and Future Work

9.1 Overview

New techniques have recently been developed to process the signals with special structures. Sparsity of signals is one of these structures. The sparse model can approximately represent a larger class of signals, called compressible signals. It is practically observed that a large class of natural signals are compressible, which facilitates the use of a sparsity model for them. A disadvantage of using this model is its high computational demand. As the computational power of computers and embedded devices have significantly been improved and multicore processing has become available, sparsity based signal processing methods have become widely accepted.

This thesis aimed to consider some less investigated aspects of sparse modeling and coding. It started by briefly introducing the problem and showing how it can be formulated as an optimization problem. Although the optimization of different objectives have mathematically been investigated for more than three centuries, suitable methods for such a difficult problem have to be selected. Some of these methods have been reviewed in Chapter 3 and a novel method, quantized iterative hard thresholding, has also been proposed for a special sparse coding problem, where quantized value representations are sought.

A linear generative model, i.e. "dictionary", is necessary in sparse approximation of signals. If the dictionary is not given, a dictionary learning method can be used to adapt an initial dictionary based on a set of training samples. These methods have briefly been reviewed in Chapter 4. A novel method was then introduced which has been shown to be an effective method for scaleable sparse approximation problems. The new method is very flexible and can be used to find structured dictionaries. Two of these structures, minimum size and compressible, were explored in the following chapters, i.e. Chapters 6 and 7.

An alternative approach to choose a dictionary is to design such a dictionary which follows certain properties. A property which facilitates the sparse coding algorithms is the incoherence of dictionary. One practical method was also presented to design such a dictionary. As the

presented algorithms here are “iterative“, a separate study on the convergence of algorithms is necessary, which is presented in the appendices B and D.

9.2 Conclusion and Future Work

This thesis has shown a potential for adaptation of the dictionary based generative model in sparse coding. As the applications of the sparse generative model, such as compressed sensing, source coding, classification, source separation and denoising, become more widely accepted, dictionary selection is receiving more attention.

Here some aspects of the dictionary selection problem have been investigated and new methods have been proposed. The new methods often perform better than current methods and have some special mathematical features. In the following, some key contributions are highlighted and some directions for the future work are given.

- *Quantized Sparse Approximations:*
  The quantized sparse approximation using iterative hard thresholding was presented in the first part of this thesis. This method can be seen as a special case of gradient-projection method where the admissible set is the set of integer value approximations. The possible advantages of using a quantized sparse approximation, were demonstraited by some simulations. The main difficulty in using such a method is that the quantizer should be defined a priori, where the optimal quantizers change in different coding rates, and the algorithm also only converges to a local minimum. An adaptive technique to change the quantizer might solve the quntization problem. The convergence analysis of this method is also left for the future work.

  Constraining the sparse approximations to being quantized introduces extra fixed points for the iterative thresholding methods, which seems to prevent the QIT method significantly outperforms a posteriori quantized sparse approximation methods.

- *Dictionary Learning:*
  The dictionary learning problem was reformulated as a bi-convex optimization problem, which was solved using a very flexible optimization technique, called the majorization minimization method. The dictionary was then learned with two update steps, dictionary and coefficients updates. As these steps are simply matrix-matrix multiplications, which

can be broken down to be done using multicore processors, followed by a very easy non-linear operator, it can be used for scalable dictionary learning. It has been compared to other dictionary learning methods and shown that it can be used with a dictionary size that is difficult, if not impossible, to use with other dictionary learning methods. Another nice feature of this method is that it is possible to analysis the convergence of the proposed algorithm, to a set of fixed points, which can be found in Appendix B. This proof distinguishes the dictionary learning using majorization methods from some of the conventional dictionary learning methods like K-SVD and MOD, that so far do not have any convergence results.

Most of the minimally constrained dictionary learning methods, like K-SVD and MOD, can be extended using some extra constraints on the dictionaries to find structured dictionaries. These structured dictionary learning methods have been surveyed in Chapter 4. The dictionary learning method proposed in this thesis can also be extended to learn structured dictionaries. Two structures had been explored here, i.e. minimum size and compressible. The minimum size structure[1] is found by inducing a joint sparsity measure, here $\ell_{1,2}$-norm, on the dictionary. It was shown in Chapter 6 that the dictionary learning method with this setting can recover an original dictionary, with a correct size. The performance of using the learned minimum size dictionary in a coding application is compared with other pre-designed dictionaries and the improvement in R-D is demonstraited. The performance of dictionary learning in this framework strongly relies on the joint sparsity penalty, θ. Here θ was selected intuitively. A systematic selection of this penalty factor is left for the future work.

The parsimonious dictionary learning framework can be extended to the dictionary design problem. The aim of parsimonious dictionary design can be to reduce the size of a pre-designed dictionary by selecting a subset of atoms in the original dictionary as the new dictionary. It is an important task, when the dictionary size has to be selected for coding at a specific coding rate. It also helps to preserve the structure of the dictionary, which is necessary for fast implementations.

The other explored structure was the compressibility of dictionary, Chapter 7. This structure lets us implement the dictionary more efficiently, if the generative model is fast, e.g. Fourier or wavelet. Constraining the dictionary to be compressible, in the domain

[1] It means learning a dictionary with the maximum number of zero atoms, which we can remove those from the final learned dictionary.

specified by Φ, biases the solutions, i.e. the learned dictionary may significantly change by choosing a different Φ. Here Φ is selected based on the prior information about the signals, which can be selected more wisely in the future work.

The compressibility model for the dictionary simplifies the dictionary learning problem by reducing the number of unknown parameters which approximately represent the dictionary. This might reduce the necessary number of training samples required for dictionary learning to prevent overfitting. An independent research on the dictionary recovery in this setting is left for the future work.

- *Parametric Dictionary Design:*

  The second half of Part II presented a new framework for dictionary design. In the new framework, the dictionary is presented using a set of parameters. The aim is to find a suitable set of parameters by which the dictionary satisfies certain constraints. Here a minimally correlated dictionary is sought. This can be formulated as an optimization problem which can approximately be solved using a relaxed alternating minimization method. The parametric dictionary design method was applied to a case study, i.e. the Gammatone dictionary. The designed dictionary was used to find sparse approximations of some audio signals. The simulation results showed an improvement in the decay rate of residual error using MP and designed dictionary. The convergence of the practical algorithm is studied in Appendix D. The parametrically designed dictionary is unstructured if the generative sets of different atoms are disjoint. A special case, where the dictionary is structured, has been investigated in [YDD10].

  An important disadvantage of the parametrically dictionary design is that the dictionary is not directly data dependent. The parametric model for the dictionary can be used in *dictionary learning* to preserve the structure of dictionary or reduce the necessary number of training samples. The dictionary would be data dependent in this setting.

- *Other Topics for Future Work:*

  This thesis introduced some new frameworks for dictionary selection and practical algorithms to find the dictionary. An important area in dictionary selection, which was not investigated here, is the theoretical study of dictionary recovery, see for example [GS08, GS09]. The aim here is to recover an original dictionary, where the permutation and sign flip of the atoms are allowed.

  Another interesting area in dictionary learning is to learn a dictionary for a specific ap-

plication, see [SV08] for classification and [RLS09] for compressed sensing as some examples. These dictionary learning methods are based on inducing extra constraints on dictionaries which facilitates the use of dictionaries for that application. An important part of these applications is compressed sensing [Don06, CRT06b]. An overcomplete dictionary can be used in this framework to sparsify the signals, just before applying a sensing matrix [RSV08]. The aim of dictionary learning for compressed sensing is to find a dictionary which sparsifies the signals and does not change the necessary mathematical properties of the overall sensing matrix, i.e. sensing matrix multiplied by the dictionary. There is currently little work done in this area [RLS09] and a more detailed investigation seems to be necessary.

Appendix A
Matrix Form of the Majorizing Function

We can use the Taylor series to majorize the quadratic term of the objective function which has a bounded curvature. The Taylor series in matrix form [Dat09, Appendix D 1.7] is given by,

$$f(\mathbf{U}) = f(\mathbf{V}) + \overset{\rightarrow \mathbf{U}-\mathbf{V}}{df}(\mathbf{V}) + \frac{1}{2!}\overset{\rightarrow \mathbf{U}-\mathbf{V}}{df^2}(\mathbf{V}) + o(||\mathbf{U}||^3) \tag{A.1}$$

where $\overset{\rightarrow \mathbf{U}-\mathbf{V}}{df}(\mathbf{V})$ and $\frac{1}{2!}\overset{\rightarrow \mathbf{U}-\mathbf{V}}{df^2}(\mathbf{V})$ are the directional first and second derivatives of f at $\mathbf{V}$ in the $\mathbf{U}-\mathbf{V}$ direction. The directional derivatives are defined by,

$$\overset{\rightarrow \mathbf{Y}}{df}(\mathbf{X}) = \{\frac{d}{dt}f(\mathbf{X}+t\mathbf{Y})\}_{t=0}, \quad \overset{\rightarrow \mathbf{Y}}{df^2}(\mathbf{X}) = \overset{\rightarrow \mathbf{Y}}{df}\,(\,\overset{\rightarrow \mathbf{Y}}{df}\,(\mathbf{X})).$$

For a bounded curvature objective function we have,

$$f(\mathbf{U}) \leq f(\mathbf{V}) + \overset{\rightarrow \mathbf{U}-\mathbf{V}}{df}(\mathbf{V}) + \frac{1}{2}tr\{(\mathbf{U}-\mathbf{V})^T\Pi(\mathbf{U}-\mathbf{V})\}, \tag{A.2}$$

where $\Upsilon = \Pi - \overset{\rightarrow \mathbf{U}-\mathbf{V}}{df^2}(\mathbf{V})$ is positive definite ($\Upsilon \succ 0$).

Appendix B

Convergence Study of the Dictionary Learning with the Majorization Minimization Method

In the first step of analyzing an iterative algorithm, we need to show the boundedness of the solutions (or the stability of the algorithm). The stability of the algorithms, in which a positive objective is reduced in each iteration, is guaranteed using Lyapunov second theorem. For example the stability of the MAP-DL is guaranteed when a *suitable* step size is chosen (to the author's knowledge, no analytical study has been done on how to choose this step size). The convergence of the alternating (gradient) projection based methods essentially depends on the admissible sets (and the gradient step size). In the dictionary learning problem with the admissible sets given by [EAH99a] [OF97], the convergence of the algorithm is not guaranteed. In K-SVD, one needs to find the sparse approximations based on the ℓ_0 sparsity measure for which no efficient algorithm exist so that the stability analysis is challenging. In practice it has been observed that in MOD and K-SVD, when the solution sequence enters a neighborhood of a local minimum, the objective increases in some iterations. Therefore, it does not converge monotonically to the solution.

The next step is to show the convergence of the algorithm to a fixed point or a set of fixed points. Kreutz Delgado *et al.* in [KMR$^+$03] referred to the convergence of the gradient flow method to show the convergence of the MAP-DL. Although this statement is completely correct, it requires the use of an arbitrary small step size which is practically impossible.

The stability of dictionary learning based on the majorization method has already been proven by the fact that we reduce the objective in each step. Here, we show the convergence to a set of fixed points. The dictionary learning using majorization minimization method can be viewed as a generalized block-relaxed minimization scheme applied to an augmented objective function.

Specifically, two majorizing objectives, (5.4) and (5.7), are combined,

$$\begin{aligned}\psi(\mathbf{D},\mathbf{X},\mathbf{D}^{\ddagger},\mathbf{X}^{\ddagger}) = \phi(\mathbf{D},\mathbf{X}) + c_D||\mathbf{D}-\mathbf{D}^{\ddagger}||_F^2 \\ + c_X||\mathbf{X}-\mathbf{X}^{\ddagger}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^{\ddagger}\mathbf{X}^{\ddagger}||_F^2\end{aligned} \tag{B.1}$$

where $\mathbf{X}^{\ddagger}$ and $\mathbf{D}^{\ddagger}$ are two auxiliary parameters corresponding to $\mathbf{X}$ and $\mathbf{D}$ respectively. c_D and c_X have been chosen to be larger than the spectral norms of $\mathbf{X}^{\ddagger T}\mathbf{X}^{\ddagger}$ and $\mathbf{D}^{\ddagger T}\mathbf{D}^{\ddagger}$ respectively. This augmented objective function *does not* majorize the joint objective, however when $(\mathbf{D},\mathbf{D}^{\ddagger}|_{\mathbf{D}^{\ddagger}=\mathbf{D}})$ or $(\mathbf{X},\mathbf{X}^{\ddagger}|_{\mathbf{X}^{\ddagger}=\mathbf{X}})$ are fixed, (B.1) majorizes the original joint objective based on the other pair of parameters. When the optimization method is viewed in the block relaxation framework, the optimum of $\mathbf{X}^{\ddagger}$ or $\mathbf{D}^{\ddagger}$ is easily found by $\mathbf{X}$ or $\mathbf{D}$ respectively. This corresponds to the parameter update in the standard majorization method [Lan04]. Therefore any sequence of updates is acceptable, given each update of $\mathbf{D}$ (or $\mathbf{X}$) is followed by an update based on $\mathbf{D}^{\ddagger}$ (or $\mathbf{X}^{\ddagger}$) respectively.

Such a block-relaxed sequential constrained minimization is not in general guaranteed to converge (see [Lee94] for some counter examples). To study the convergence of the dictionary learning with the majorization minimization method, we need to do a little more work. In the next subsection, some theoretical analysis of the generalized block relaxation method will be introduced. The proposed algorithm will then be analyzed for dictionary learning, based on the given theoretical analysis.

B.1 Generalized block relaxed iterative mappings and their convergence

Let $\eta(\omega) : \Omega \rightarrow \mathbb{R}$ be the multiparameter objective function which we want to minimize. Let Υ be the set of admissible parameters. The parameter ω is defined as the concatenation of the blocks of parameters $\{\omega \in \Upsilon :\ \omega = (\omega_1, \omega_2, ..., \omega_p)\ , \omega_i \in \Omega_i\}$ where $\Omega = \Omega_1 \times \Omega_2 \times ... \times \Omega_p$. In dictionary learning based on block relaxation, $p = 2$, $\omega_1 = \mathbf{X}$ and $\omega_2 = \mathbf{D}$. In generalized block-relaxed dictionary learning, $p = 4$ as we have two more auxiliary parameters $\mathbf{X}^{\ddagger}$ and $\mathbf{D}^{\ddagger}$.

Now it is needed to introduce point to set maps,

***Definition* B.1.1** (Point to set map)**.** Let Υ be an arbitrary set and let Γ be the set of all subsets of Υ. A map $\Delta : \Upsilon \rightarrow \Gamma$ is a point to set map (see for example [Zan69]).

In the block relaxation technique a set of point to set maps $\Delta_i : \Upsilon \rightarrow \Gamma$ are defined as $\Delta_i(\widehat{\omega}) = \{\omega \in \Upsilon : \forall j \neq i \; \omega_j = \widehat{\omega}_j\}$ where $\widehat{\omega} = (\widehat{\omega}_1, \widehat{\omega}_2, ..., \widehat{\omega}_p)$ is the current value of the parameters. These point to set maps keep all the blocks of parameters fixed apart from the i^{th} block.

By starting from $\omega^{[0]}$, the set of possible solutions Λ in the minimization problem is defined as, $\Lambda = \{\omega \in \Upsilon : \eta(\omega) \leq \eta(\omega^{[0]})\}$. For any $\omega \in \Lambda$ in each block update we minimize the objective for the selected parameters. This gives us the following updating operator:

$$U_i : \Lambda \rightarrow \{u \in \Delta_i(\widehat{\omega}) : \eta(u) \leq \eta(t), \forall t \in \Delta_i(\widehat{\omega})\} \tag{B.2}$$

In general this updating operator is a point to set map and we can choose an update parameter within the resulting set. In our case, the objective function always has a unique minimizer and the updating operators are point-to-point mappings. To use a set of updating operators, we also need to have an operator selector.

***Definition* B.1.2** (Operator selector)**.** $s(k) : \mathbb{N} \rightarrow \mathscr{P}$ which $\mathscr{P} = \{i : 1 \leq i \leq p\}$

This operator can choose the updating operator by sequentially selecting (circular) or free steering through the available operators. By using the updating operators defined in (B.2) and an update selector $s(k)$, we can summarize the (generalized) block relaxed minimization by the following algorithm,

***Algorithm* B.1.1.** *Let $\omega^{[0]}$ be a given starting point, then $\{\omega^{[k]}\}_{k\in\mathbb{N}}$ is the sequence of updates given by $\omega^{[k+1]} \in U_{s(k)}\{\omega^{[k]}\}$ and stop when $\forall i \in \mathscr{P} : \widehat{\omega} = U_i\{\widehat{\omega}\}$*

When the updating operator is injective, $\omega^{[k+1]} = U_{s(k)}\{\omega^{[k]}\}$, to analyze the sequence generated by Algorithm B.1.1, we need to introduce some characteristics of the infinite series.

***Definition* B.1.3** (Asymptotically regularity)**.** A sequence $\{\alpha^{[n]}\}_{n\in\mathbb{N}}$ is asymptotically regular if $||\alpha^{[n+1]} - \alpha^{[n]}|| \rightarrow 0$, when $n \rightarrow \infty$.

$||\,.\,||$ is a norm defined in the solution space. An operator is called asymptotically regular when the series generated by the sequential use of that operator is asymptotically regular.

***Definition* B.1.4** (Essentially periodic)**.** An infinite sequence $\{\alpha^{[n]}\}_{n\in\mathbb{N}}$ drawn from a finite alphabet $\mathscr{P} = \{\mathscr{A}_i : 1 \leq i \leq p\}$ is essentially periodic, with a period $m \in \mathbb{N}, m \geq p$ when $\forall j \in \mathbb{N}, \forall \mathscr{A}_i \in \mathscr{P}, \exists n \in [jm + 1, (j + 1)m]$ and $\alpha^{[n]} = \mathscr{A}_i$.

The sequence of $\{\omega^{[k]}\}$ of the Algorithm B.1.1 is asymptotically regular when Δ_i and η satisfy the following hypotheses [FH79],

***Hypotheses* B.1.1.** For all $i \in \mathscr{P}$ and $\eta : \Upsilon \to \mathbb{R}$,

- $\forall \omega : \omega \in \Delta_i(\omega)$
- Δ_i is continuous on Υ
- $\forall \omega \in \Upsilon$, η has a unique minimizer over $\Delta_i(\omega)$
- $\exists \omega^{[0]} \in \Upsilon$ such that Λ is a compact subset.

The accumulation points of Algorithm B.1.1, when the Hypotheses B.1.1 are satisfied, can now be studied. From basic mathematical analysis, we know that any bounded sequence has at least one accumulation point (Bolzano-Weierstrass Theorem [Pal91, Theorem 4.1]). As Λ is closed, the accumulation points of $\{\omega^{[n]}\}$ are in Λ.

***Theorem* B.1.1.** *[FH79, Theorem 15] Let the update selector, $s(k)$, be essentially periodic and Δ_i and η satisfy Hypotheses B.1.1. Every accumulation point ω^* of $\{\omega^{[n]}\}$, generated by Algorithm B.1.1, satisfies $\omega^* = U_i\{\omega^*\}$ for any $i \in \mathscr{P}$*

The set of accumulation points T belongs to a level set of η. If η is continuous, T is closed and as Λ is bounded and $T \subseteq \Lambda$, T is bounded. Therefore T is compact.

***Proposition* B.1.1.** [Lan04, Proposition 10.3.1] If a bounded sequence $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ is asymptotically regular, then its set of accumulation points is connected. If this set is finite, then it reduces to a single point.

In a normed space, the following lemma guarantees that the sequence $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ generated by Algorithm B.1.1 will stay arbitrarily close to the accumulation points, when $n > N$ for some N.

***Lemma* B.1.1.** *Let $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ be a bounded asymptotically regular sequence and T be the set of its accumulation points then, $\forall \epsilon > 0, \exists N \in \mathbb{N}, for\ n > N, \exists t \in T, ||\omega^{[n]} - t|| < \epsilon$*

Proof. Let S be an ϵ-neighborhood of T and S_c be its complement in the admissible set. As the admissible set is compact, S_c is also compact. Because S is a neighborhood of T there is no

accumulation point t in S_c. If $\{\omega^{[n]}\}$ has infinitely many points in S_c, then it has a converging subsequence and at least one accumulation point in S_c . This contradicts the fact that there is no accumulation point in S_c. Therefore $\exists N : \omega^{[n]} \in S,\ \forall n > N$. On the other hand ϵ-neighborhood implies that for all $n > N$, $\exists t \in T : ||\omega^{[n]} - t|| < \epsilon$. □

In the next subsubsection, asymptotic regularity of the generalized block relaxation method for dictionary learning will be shown. This is followed by showing the convergence of the proposed method to a set of fixed points.

B.2 Convergence study of the generalized block relaxed dictionary learning

In dictionary learning, there are two parameters, coefficient matrix and dictionary. In generalized block-relaxed dictionary learning (B.1), we have four parameters. The augmented function (B.1) majorizes (5.1) only when one pair of parameter blocks ($(\mathbf{D}, \mathbf{D}^\ddagger|_{\mathbf{D}^\ddagger=\mathbf{D}})$ or $(\mathbf{X}, \mathbf{X}^\ddagger|_{\mathbf{X}^\ddagger=\mathbf{X}})$) is fixed. Therefore $\Delta_{\mathscr{X}} : \mathscr{X} \in \{\mathbf{D},\ \mathbf{X},\ \mathbf{D}^\ddagger,\ \mathbf{X}^\ddagger\}$ are the point to set maps which fix all parameters but $\mathscr{X}$ (from now on, this indexing for the point to set maps will be used).

***Proposition* B.2.1.** The generalized block-relaxed minimization of (B.1) is asymptotically regular when the updates of $\mathbf{D}$ and $\mathbf{X}$ are followed by updating of $\mathbf{D}^\ddagger$ and $\mathbf{X}^\ddagger$ respectively.

Proof. To show the asymptotic regularity we show that all the hypotheses in Hypotheses B.1.1 are satisfied. $\Delta_{\mathscr{X}} : \mathscr{X} \in \{\mathbf{D},\ \mathbf{X},\ \mathbf{D}^\ddagger,\ \mathbf{X}^\ddagger\}$ are self contained, i.e. $\widehat{\mathscr{X}} \in \Delta_{\mathscr{X}}\{\widehat{\mathscr{X}}\}$, and continuous. Therefore they satisfy the first two hypotheses. The minimum of (B.1) based on each parameter is unique (the sparse approximation minimum is reached using soft shrinkage (5.6) over $\mathbf{A}$ and the dictionary update is reached by one of the operators introduced in (5.13), (5.20) or (6.7) over $\mathbf{B}$). (B.1) is strictly convex based on $\mathbf{X}^\ddagger$ or $\mathbf{D}^\ddagger$ when all other parameters are fixed. Therefore minimization based on $\mathbf{D}^\ddagger$ or $\mathbf{X}^\ddagger$ has a unique solution. Surrogate objective function (B.1) is a continuous function. When a mapping is continuous, its epigraph Λ is a closed set [Roc70, Theorem7.1]. As the admissible set is a closed set, the intersection of Λ and this set, which is the possible solution set, is closed. On the other hand there is no infinitely large point in Λ (maximum value of $||\mathbf{D}||_F$ and $J_{1,1}(\mathbf{X})$ are bounded based on the dictionary constraints and $\phi(\mathbf{D}^{[0]}, \mathbf{X}^{[0]})/\lambda$ respectively). In an Euclidean space boundedness and closeness

are sufficient for a set to be compact. Therefore the hypothesis is satisfied and the sequence of $(\mathbf{D}, \mathbf{X}, \mathbf{D}^{\ddagger}, \mathbf{X}^{\ddagger})^{[i]} : i \in \mathbb{N}$ is asymptotically regular [FH79]. □

Finally we present a Proposition which shows the convergence of the proposed algorithm.

***Proposition* B.2.2.** Generalized block-relaxed dictionary learning converges to a single fixed point $(\mathbf{D}^*, \mathbf{X}^*)$ or gets arbitrary close to a continuum of accumulation points, where each accumulation point satisfies:

- $\psi(\mathbf{D}^*, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) \leq \psi(\mathbf{D}^*, \mathbf{X}, \mathbf{D}^*, \mathbf{X}^*) : \forall \mathbf{X}$
- $\psi(\mathbf{D}^*, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) \leq \psi(\mathbf{D}, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) : \forall \mathbf{D} \in \mathscr{D}$

Proof. Due to Proposition B.2.1, the sequence generated by generalized block-relaxed dictionary learning is asymptotically regular. Due to Theorem B.1.1 and Lemma B.1.1, the algorithm converges either to a fixed point or gets arbitrary close to a continuum of accumulation points. Because any accumulation point of the algorithm is a fixed point for all $U_i : \forall i \in \mathscr{P}$ [FH79, Theorem 15], $\mathbf{X}^*$ is the best coefficient matrix using dictionary $\mathbf{D}^*$ and $\mathbf{D}^*$ is the best admissible dictionary, using $\mathbf{X}^*$ as the sparse representation. □

Appendix C
Derivation of the Gammatone Dictionary Gradient

The gradient of the parametric Gammatone dictionary with the generative function (8.13) is calculated in this appendix. Let $\mathbf{D}_\Gamma \in \mathbb{R}^{d\times N}$ and $\Gamma \in \mathbb{R}^{4\times N}$. The i^{th} column of $\mathbf{D}_\Gamma$ is a function of the i^{th} column of Γ, $\mathbf{d}_{\gamma_i}$. The rank of $\nabla_\Gamma \mathbf{D}_\Gamma$ is 3 and it is represented by a tensor in $\mathbb{R}^{4\times d\times N}$. Each sub-matrix of this tensor (fixing the third index) is the gradient of the corresponding atom in $\mathbf{D}_\Gamma$. Therefore it is only needed to calculate the gradient of $\mathbf{d}_{\gamma_i}$ based on γ_i. Because $\mathbf{d}_\gamma$ is calculated using (8.16), a formulaton for the gradients of $g_\gamma(t)$ based on t_c, f_c, n and b is needed which can be found by the following formulas,

$$\begin{aligned}
\frac{\partial g_\gamma}{\partial t_c} &= -a((n-1)t_s^{n-2}\cos 2\pi f_c t_s + 2\pi b B t_s^{n-1}\cos 2\pi f_c t_s \\
&\qquad + 2\pi f_c t_s^{n-1}\sin(2\pi f_c t_s))e^{-2\pi b B t_s}, \\
\frac{\partial g_\gamma}{\partial f_c} &= a t_s^{n-1}(-2\pi t_s \frac{dB}{df_c}\cos(2\pi f_c t_s), \\
&\qquad - 2\pi t_s \sin(2\pi f_c t_s))e^{-2\pi b B t_s}, \\
\frac{\partial g_\gamma}{\partial n} &= a\ln(t_s)\, t_s^{n-1} e^{-2\pi b B t_s}\cos(2\pi f_c t_s), \\
\frac{\partial g_\gamma}{\partial b} &= -2\pi a B t_s^n e^{2\pi b B t_s}\cos(2\pi f_c t_s),
\end{aligned} \tag{C.1}$$

where $t_s = t - t_c$ and $\frac{dB}{df_c} = 1/Q$. The final step of calculating $\nabla_\Gamma \mathbf{D}_\Gamma$ is done by sampling the parameter t.

Some researchers have proposed more complex formulations for B. In this case, one can substitute B and $\frac{dB}{df_c}$ in (C.1) to find the gradient.

Appendix D
Convergence Study of the Parametric Dictionary Design

In this appendix, it is first shown that if $\mathbf{D}_\Gamma$ is a differentiable function on Υ, the Algorithm 5 reduces the proposed objective function in (8.8). It is then shown that when Υ is compact[1], the sequence generated by the algorithm, gets as close as possible to a set of limit points.

***Proposition* D.0.3.** Let $\mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$ be the Gram matrix of the dictionary at k^{th} iteration. The Algorithm 5 reduces, or remains the same, $\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F$ in each update of the parameters ($\Gamma_k \rightarrow \Gamma_{k+1}$), where $\mathcal{P}_{\Lambda^N}$ is the orthogonal projector onto Λ^N.

Proof. Let $\mathbf{G}_{P_{k+1}}$ be an abbreviation for $\mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}$, which is found using (8.9). Using the parameter update step (line 5) and the fact that $\mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$,

$$\begin{aligned}
\alpha_k \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F &= \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F \\
&\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{R_{k+1}}\|_F \\
&= \|\mathbf{G}_{\Gamma_{k+1}} - \alpha_k \mathbf{G}_{P_{k+1}} - (1-\alpha_k)\mathbf{G}_{\Gamma_k}\|_F \\
&= \|(\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}) - (1-\alpha_k)(\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}})\|_F \\
&\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}\|_F - (1-\alpha_k)\|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F \ ,
\end{aligned}$$

where the triangular inequality has been used to derive the last inequality. This provide the following inequality,

$$\begin{aligned}
\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F &\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}\|_F \\
&\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_{k+1}}\|_F,
\end{aligned} \tag{D.1}$$

where the minimum distance property of the orthogonal projection is used in the last inequality.

□

[1] In Euclidean space, a set is compact if it is closed and bounded.

Let Υ° be the interior of Υ and $\operatorname{class} C^1$ be the set of all (at least) one time differentiable functions. To show the convergence of Algorithm 5 to a set of fixed points, the following two Lemmata are needed.

***Lemma* D.0.1.** *Let $\mathbf{D}_\Gamma : \Upsilon^\circ \to \mathbb{R}^{d\times N} \in \operatorname{class} C^1$ and Υ be compact. The epigraph of the objective (8.8) at an admissible Γ_0 is compact.*

Proof. When the parametric dictionary $\mathbf{D}_\Gamma$ is differentiable on Υ, the objective function in (8.8) is continuous with respect to Γ. Let $\Gamma^* \in \Upsilon$ and $\mathbf{G}^* \in \Lambda^N$ be the solution of (8.8) and $\mathbf{G}^*_\Gamma = \mathbf{D}^T_{\Gamma^*}\mathbf{D}_{\Gamma^*}$. Then $\mathbf{G}^* = \mathcal{P}_{\Lambda^N}\mathbf{G}^*_\Gamma$. Therefore the objective in (8.8) can be replaced by the following surrogate objective based on Γ as the only parameter,

$$(\sum_{i\neq j}(|\{g_\Gamma\}_{i,j}| - \mu_G)^2 + \sum_{i=j}(\{g_\Gamma\}_{i,j} - 1)^2)^{1/2}, \tag{D.2}$$

where $|\{g_\Gamma\}_{i,j}|$ is the absolute value of the (i,j) element of $\mathbf{G}_\Gamma = \mathbf{D}^T_\Gamma\mathbf{D}_\Gamma$. This objective function is a continuous function of Γ. The continuity of the objective function and the compactness of Υ prove the compactness of the epigraph of the objective at an admissible point Γ_0 [BV04]. □

Due to the Bolzano-Weierstrass theorem, Algorithm 5 has a set of accumulation points. The Lemma B.1.1 is now reformulated for a more general (including asymptotically non-regular[2]) sequence. Although the proof is the same, the set of accumulation points can be dis-connected, when the sequence is not asymptotically regular.

***Lemma* D.0.2.** *Let $\{\Gamma_n\}_{n\in\mathbb{N}}$ be an infinite sequence in a compact set and T be the set of its accumulation points then, $\forall\epsilon > 0, \exists N \in \mathbb{N}$, such that for $n > N, \exists\Gamma^\ddagger \in T, \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$*

Proof. Let S be an ϵ-neighburhood of T and S_c be its complement in the admissible set. As the admissible set is compact, S_c is also compact. Because S is a neighburhood of T there is no accumulation point Γ in S_c. If $\{\Gamma_n\}$ has infinite many points in S_c, then it has a converging subsequence and at least one accumulation point in S_c. This contradicts the fact that there is no accumulation point in S_c. Therefore $\exists N : \Gamma_n \in S,\ \forall n > N$. On the other hand ϵ-neighburhood implies that for all $n > N$, $\exists\Gamma^\ddagger \in T : \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$. □

[2] A sequence $\{a_k\}_{k\in\mathbb{N}}$ in a normed space is called asymptotically regular when $\lim_{k\to\infty}\|a_k - a_{k-1}\| = 0$

***Theorem* D.0.1.** *Let* $\mathbf{D}_\Gamma \in \operatorname{class} C^1$. *The Algorithm 5 converges to a set of fixed points by starting from* $\Gamma_0 \in \Upsilon$, *where* Υ *is a compact set.*

Proof. Due to Lemma D.0.1 the epigraph of the objective in (8.8) at Γ_0 is compact. The Proposition D.0.3 shows that the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ is in this epigraph. The convergence of the Algorithm 5 to a non-empty set of accumulation points is guaranteed using Lemma D.0.2. Line 5 of Algorithm 5 prevents the existence of a continuum of accumulation points. Therefore, the accumulation points are fixed points. □

Appendix E
Publications

Peer Reviewed Journal Articles:

1. “Parametric Dictionary Design for Sparse Coding”, with L. Daudet, M. Davies, IEEE Transaction on Signal Processing, Vol. 57, No. 12, pp 4800-4810, 2009.

2. “Dictionary Learning for Sparse Approximations with the Majorization Method”, with T. Blumensath, M. Davies, IEEE Transaction on Signal Processing, Vol. 57, No. 6, pp 2178-2191, 2009.

Conference Proceedings:

1. “Structured and Incoherent Parametric Dictionary Design”, with L. Daudet and M. Davies, accepted for presentation in the IEEE International Conference on Acoustics, Speech and Signal Processing, 2010 (Invited Paper).

2. “Compressible Dictionary Learning for Fast Sparse Approximation”, with M. Davies, IEEE Workshop on Statistical Signal Processing, 662-665, Aug. 31- Sept. 3, 2009.

3. “Parsimonious Dictionary Learning”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing,pp. 2869-2872, April 2009.

4. “Parametric Dictionary Design for Sparse Coding”, with L. Daudet, M. Davies, Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09), 2009.

5. “Regularized Dictionary Learning for Sparse Approximation”, with T. Blumensath, M. Davies, European Signal Processing Conference (EUSIPCO), August 2008.

6. “Iterative Hard Thresholding and L_0 Regularisation”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing, 877-880, April 2007.

7. “Quantized Sparse Approximation with Iterative Thresholding for Audio Coding”, with T. Blumensath, M. Davies, IEEE International Conference on Acoustics, Speech and Signal Processing, 257-260, April 2007.

QUANTIZED SPARSE APPROXIMATION WITH ITERATIVE THRESHOLDING FOR AUDIO CODING

M. Yaghoobi, T. Blumensath and M. Davies

Institute for Digital Communications,
Joint Research Institute for Signal and Image Processing,
University of Edinburgh, UK

ABSTRACT

Sparse coding is a new field in signal processing with possible applications to source coding. In this paper we present a new method that combines the problems of sparse signal approximation with coefficient quantization. This method uses overcomplete dictionaries and exploits signal redundancy. The proposed method will be derived as an extension of a recently presented method (iterative thresholding) to find sparse representations of signals. Because in digital communication and storage we need a quantized representation of the signal, instead of quantization of sparse representations a posteriori, we propose a refined method that combines sparse approximation and quantization. To compare the proposed method to a posteriori quantization, we present an audio example.

Index Terms **–** Sparse approximation, Quantization, Iterative Thresholding, Audio coding, Signal representation

1. INTRODUCTION

Sparse approximations represent signals with a small number of elementary functions (atoms) from an overcomplete set of functions (dictionary). This kind of signal representation has various applications such as source separation, denoising, feature extraction, compression and source coding. The focus of this paper is to simultaneously obtain a sparse and quantized representation of a signal. As an example, we use an audio signal to show the performance of the algorithm. Sparse representations are potentially useful in source coding because the encoder only needs to encode non-zero coefficients and their indices (i.e. the indices of the atoms in the dictionary) to enable the decoder to reconstruct the original signal.

Most modern audio codecs use a transformation of the input as the first step to get a sparser representation of the signal and, with some psychoacoustic considerations, quantize and encode the coefficients. The decoder uses the inverse transform [1]. The idea behind transform coding is that a simlpe scalar quantizer can be used. Therefore, many researchers use sparse representations based on overcomplete dictionaries to increase the sparsity of the representation (with an increase in the cost of index coding)[2] [3] [4].

Previous approaches mostly use greedy algorithms like Matching Pursuit (MP) or its extension, Quantized MP (QMP) [5], which was shown to improve quantized SNR by 0.5–2 dBs for a fixed bit rate [6]. In this paper we propose a different in-loop quantization method and show that it uses the redundancy in the dictionary to find a better quantized approximation. The contribution of this paper is an iterative algorithm that jointly optimizes the selection of atoms from a redundant dictionary and the quantization. A new penalty function will be presented to replace the traditional penalty function based solely on the number of none-zero coefficients. To optimize this penalty function we need either relaxation or approximation. In this work the latter one is chosen.

2. SPARSE APPROXIMATION AND ITERATIVE THRESHOLDING

An optimal source code can be achieved by Vector Quantization (VQ) [7] which is computationally expensive. Transform coding is used to get suboptimal source codes with simpler algorithms. In standard transform coding, coefficients are quantized with a scaler quantizer and then entropy coded [8]. Linear transforms do not always lead to good performance. One solution is to represent the signals using a nonlinear transform and an overcomplete set of elementary functions. Nonlinear transforms can lead to sparser representations for coding. Overcomplete signal representations can be formulated as,

$$y = Kx, \tag{1}$$

where K is an N by M matrix with $M > N$ and $|K| = N$. y and x are the input signal and the signal in the transform domain. Because K is a non-square matrix with $M > N$, we have an infinite number of solutions x for every input y. We can choose a particular solution based on the constrained

This work is funded by EPSRC grant number D000246.The authors acknowledge their support of the Joint Research Institute with the Heriot-Watt University as a component part of Edinburgh Research Partnership.

optimization of the desired penalty function $P(x)$,

$$\min_{x:y=Kx} P(x) \tag{2}$$

For sparse representations, $P(x)$ is often chosen to be l_0, which measures the number of non-zero coefficients. Instead of solving this exact representation problem, we use an additive cost function of a squared error approximation and the penalty,

$$\min_x \Phi(x) \quad ; \quad \Phi(x) = ||Kx - y||^2 + \lambda P(x) \tag{3}$$

where $||\cdot||$ is the norm in signal space and λ a Lagrangian multiplier. In general, solving the above optimization problem based on the l^0 sparsity constraint is an NP-hard problem and is not computable in an acceptable amount of time. So the problem needs to be sinplified using relaxation or approximation [9]

Recently Daubechies et al. [10] have presented an Iterative Thresholding algorithm (IT), as an iterated version of classical thresholding [11] to find sparse approximations for a broader ranges of dictionaries (the classical one was presented for orthogonal wavelets and could be extended to other orthogonal bases). The algorithm was shown to solve a relaxed version of the l^0 problem (with a convex penalty function). The penalty function in [10] is,

$$P(x) = |x|_p^p \tag{4}$$

where $|x|_p$ is the p-norm with $1 \leq p \leq 2$ to ensure convexity of $P(x)$.

The matrix K couples the coefficients and prevents us from optimizing the cost function element-wise. This coupling can be removed by adding a convex function to the cost function, to get a "surrogate function". We can then optimize the new cost function (this process is called optimization transfer),

$$\Phi^S(x, x') = \Phi(x) + ||x - x'||^2 - ||Kx - Kx'||^2 \tag{5}$$

When $x = x'$, the surrogate function is equal to the original cost function. Rewriting (5) yields,

$$\Phi^S(x, x') = \sum_i [(x_i - \alpha_i)^2 + \lambda |x_i|^p] + [\beta - \alpha_i^2]. \tag{6}$$

where $\alpha = (I - K^*K)x' + K^*y$, $\beta = ||y||^2 + ||x'||^2 - ||Kx'||^2$, i shows the element number and K^* is the conjugate transpose of K. α is a function of x also known as a Landweber update of x [12], which could be used iteratively to compute the l^2 regularized optimal solution of the inverse problem. The second term is constant and we only need to optimize the first sum, which is now decoupled and can be minimized elementwise. In an iterative scheme we set the previous computed value, x^{n-1}, to x' and then set x^n to the value x that optimizes

$$\Phi^S(x^n, x^{n-1}, i) = (x_i^n - \alpha_i^{n-1})^2 + \lambda |x_i^n|^p \tag{7}$$

where α^{n-1} is the Landweber update of x^{n-1}. The convergence of this algorithm to a minimum of (3), for certain cost functions, is shown in [10]. In each step we find the best value for x_i^n based on x_i^{n-1} (or its corresponding Landweber update). Therefore the iterative algorithm for M iterations is as follows:

1. $n = 1$, $x^0 = 0$,
2. $\alpha^{n-1} = (I - K^*K)x^{n-1} + K^*y$,
3. $x_i^n = f(\alpha_i^{n-1}); \forall i$
4. $n = n + 1$ if $n \leq M$ return to step 2.

In step 3, f is the element-wise optimizer. When $p = 1$ and $p = 0$ this function is soft- and hard- thresholding [11], respectively.

The IT algorithm is flexible and it is possible to change the penalty function (albeit under certain conditions). In this paper we propose a Quantized IT algorithm based on certain modifications of the cost function, such that we simultaneously get a quantized signal representation.

3. QUANTIZED SPARSE APPROXIMATION

In this section we are considering the problem of quantized sparse representations. For coding, coefficients need to be quantized. Therefore the transform is changed to get quantized coefficients to reduce quantization error. The quantized version of (3) is:

$$\Phi_Q(z) = ||Kz - y||^2 + P_Q(z) \tag{8}$$

$P_Q(z) = \lambda ||z||_0$ measures the number of non-zero coefficients and z is a quantized value vector with the desired uniform quantizer, with larger zero bin (δ_0 and δ_1 are the zero and non-zero bin sizes). Optimizing the above cost function is an NP-hard problem. But with iterative thresholding in the quantized domain we could decrease this cost function progressively. After adding quantized version of the previously mentioned convex function, the following surrogate function should be minimized in each step:

$$\Phi^S(z^n, z^{n-1}, i) = (z_i^n - \alpha_i^{n-1})^2 + \lambda |z_i|^0 \tag{9}$$

Here $|z_i|^0$ is equal to zero if $z_i = 0$ and equal to one otherwise. We are looking for the optimum value of Φ^S in the quantized value domain. For different z_i^n, Φ^S is

$$\Phi^S(z^n, z^{n-1}, i) = \begin{cases} (\alpha_i^{n-1})^2 & z_i^n = q_0 = 0 \\ (\alpha_i^{n-1} - q_k)^2 + \lambda & z_i^n = q_{k:k \neq 0} \end{cases} \tag{10}$$

where q_k is the k^{th} quantization level ($k \in Z, -\lfloor L/2 \rfloor + 1 \leq k \leq \lfloor L/2 \rfloor$ for a L level quantizer). To define the neighborhood of each q_k in which the optimum value of Φ^S (for the quantized value z_i^n) is q_k , we just need to compare it with Φ^S at adjacent quantization point(s) (q_{k-1} and q_{k+1}). This leads

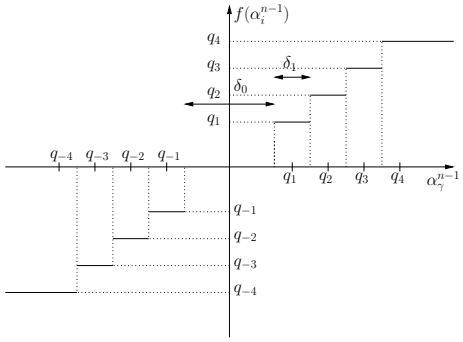

Fig. 1. 9 level on-center QShrinker

to a function on α that is a quantizer with the same quantization levels as the original quantization levels and an adjustable zero bin. We can choose an appropriate λ, by using equation (11), to ensure that the quantizer is uniform in non-zero bins and has a larger zero-bin size, see Figure 1.

$$\lambda = (\delta_0/2)^2 - (\delta_1/2)^2 \tag{11}$$

Therefore the shrinking function changes to a simple uniform quantizer,

$$f(\alpha) = Q(\alpha) \tag{12}$$

With different initial values, the algorithm will converge to different fixed points. Increasing the number of quantization levels directly increase the number of local minima. To improve performance, we adopt a relaxation strategy for iterative shrinkage previously presented in [13]. Instead of updating the current coefficients with the proposed threshold, we choose a relaxation factor μ and update the current coefficients as

$$x_i^n = (1-\mu)x_i^{n-1} + \mu f(\alpha_i^{n-1}), \tag{13}$$

where $0 < \mu \leq 1$. With this update, x_i is not quantized. But it is obvious that the fix points of (12) and (13) are the same. After the algorithm converges, all x_is have quantized values. When $||K|| > 1$, for some initial values, updating by (12) is unstable. But with the use of this relaxation, and choosing appropriate μ, our simulations show stability for both methods (IT and QIT). The overall process is the same as IT but with step 3 replaced by (13).

4. SIMULATIONS

A segment of pop music sampled at 32kHz was chosen here as a test signal (Figure 2). A 4 times overcomplete MDCT dictionary (overcomplete in the frequency domain) was used.

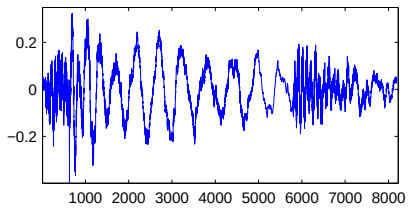

Fig. 2. Input audio signal

All simulations were started with $x^0 = 0$. We fixed quantization levels and used a uniform dead-zone quantizer with the following zero bin to non-zero bin ratio,

$$\zeta = \frac{\delta_0}{\delta_1} \tag{14}$$

By changing ζ, the results of the algorithm will have a varying umber of non-zero coefficients (it should be noted that this convention is not just for QIT. It is also used for IT, where the zero bin is the thresholding parameter. So we can compare equivalent coefficients quantized with QIT for a specific number of non-zero coefficients). A four bit quantizer (16 levels) was selected to quantize each coefficient. To show the convergence of the algorithm, simulations were run for 20 and 100 iterations. The results are shown in Figure 3. The graph with plus symbols is iterative hard thresholding and the results achieved when quantizing this solution are shown with circles. QIT and its quantized output are shown with cross and star symbols. Note that due to the relaxation approach used, the output of QIT is not automatically quantized. The horizontal axis shows the number of non-zero coefficients. We can see that for differenet numbers of non-zero coefficients, IT gives better SNR than QIT. However after quantization of the coefficients, the SNR of the decoded quantized coefficients of QIT is better than quantized IT. We also see that with more iterations, QIT and its quantized output get closer to each other, which shows that the algorithm is converging to a quantized solution. Another observation to be made here is that the SNR starts to decrease when we use a large number of non-zero coefficients. This is an artifact in the analysis where we use a fixed coding cost, i.e. a fix number of quantization levels. To show the benefit of using QIT, we need to show the operating rate-distortion (R-D) curve by computing the convex hull for different bit budgets. The audio sample used in the previous experiment is here used for coding with 4 to 9 bit quantizers. The operational R-D is shown in Figure 4. The graph shows that we have 0.2 dB SNR improvement for 1 bit/sample and up to 1 dB improvement for 12 bits/sample.

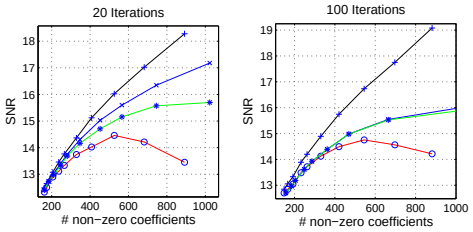

Fig. 3. For two different numbers of iterations (20 and 100) output SNRs are shown in four different cases (IT (+), QIT (x), quantized QIT (*), quantized IT (o))

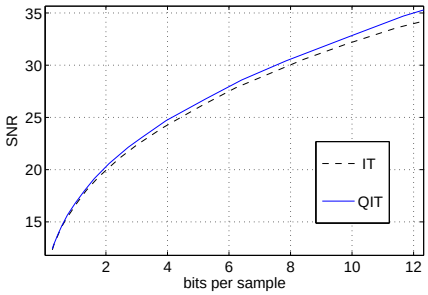

Fig. 4. Operating R-D curves for QIT (upper) and IT (lower)

5. CONCLUSION

In this paper we introduced a new method for jointly approximating and quantizing a signal. The newly presented iterative thresholding method was refined for this purpose and we have shown that even with a small number of iterations the algorithm can give a relatively good result (close to the fixed point). The algorithm is much faster than previously used MP type algorithms. Each iteration of MP and QIT have the same order of computation. However MP extracts one element at a time and therefore requires at least as many iterations as the number of atoms to be extracted, while QIT calculates all the coefficients with less iterations. With the proposed method, we have shown that jointly quantized and sparsified coefficients achieve a better SNR for the same number of non-zero coefficients than sparse approximation and quantization done separately. Because a psychoacoustic model was not considered, this kind of coefficient coding is not comparable with some well known available coders. This paper aims to show the preference of using quantized sparse approximation instead of a posteriori quantization of sparse representation. More investigations are required to study ways of choosing the relaxation parameter, finding an appropriate initialization, considering psychoacoustic models and using listening test for final evaluation.

6. REFERENCES

[1] M. Bosi, *Introduction to Digital Audio Coding and Standards*, Springer, 2002.

[2] M.M. Goodwin, *Adaptive Signal Models: Theory, Algorithms and Audio Applications*, Ph.D. thesis, University of California, Berkeley, 1997.

[3] P. Frossard, P. Vandergheynst, R.M. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," *IEEE Trans. on Signal Processing*, vol. 52, no. 2, pp. 525–535, 2004.

[4] M. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, pp. 457–470, 2006.

[5] V.K. Goyal, M. Vetterli, and N.T. Thao, "Quantized overcomplete expansions in R^N : Analysis, synthesis and algorithms," *IEEE Trans. on Information Theory*, vol. 44, no. 1, pp. 16–31, 1998.

[6] C.D. Vleeschouwer and A. Zakhor, "In-loop atom modulus quantization for matching pursuit and its application to video coding," *IEEE Trans. on Image Processing*, vol. 12, no. 10, pp. 1226–1242, 2003.

[7] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.

[8] V.K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, 2001.

[9] J.A. Tropp, *Topics in Sparse Approximation*, Ph.D. thesis, University of Texas, Austin, 2004.

[10] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math*, vol. 57, pp. 1413–1541, 2004.

[11] D.L. Donoho and I.M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[12] L. Landweber, "An iterative formula for Fredholm integral equations of the first kind," *Americam Journal of Mathematics*, vol. 73, pp. 615–624, 1951.

[13] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," to appear in IEEE Trans. on Information Theory.

REGULARIZED DICTIONARY LEARNING FOR SPARSE APPROXIMATION

M. Yaghoobi, T. Blumensath, M. Davies

Institute for Digital Communications,
Joint Research Institute for Signal and Image Processing,
University of Edinburgh, UK

ABSTRACT

Sparse signal models approximate signals using a small number of elements from a large set of vectors, called a dictionary. The success of such methods relies on the dictionary fitting the signal structure. Therefore, the dictionary has to be designed to fit the signal class of interest. This paper uses a general formulation that allows the dictionary to be learned form the data with some *a priori* information about the dictionary. In this formulation a universal cost function is proposed and practical algorithms are presented to minimize this cost under different constraints on the dictionary. The proposed methods are compared with previous approaches using synthetic and real data. Simulations highlight the advantages of the proposed methods over other currently available dictionary learning strategies.

1. INTRODUCTION

Signals can be approximated using overcomplete representations with more elementary functions (atoms) than the dimension of the signal. These representations are not unique for a given set of atoms. A sparse representation is an overcomplete representation that uses the minimal number of non-zero coefficients. For example, sparse representations have been used for low bitrate coding, denoising and source separation. Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ (where $d < N$) be the input and the coefficient vectors and let the matrix $\mathbf{D} \in \mathbb{R}^{d \times N}$ be the *dictionary*. One form of sparse approximation is to solve an unconstrained optimization problem,

$$\min_{\mathbf{x}} \Phi(\mathbf{x}) \; ; \; \Phi(\mathbf{x}) = ||\mathbf{y} - \mathbf{D}\mathbf{x}||^2 + \lambda ||\mathbf{x}||_0 \qquad (1)$$

where $||\mathbf{x}||_0$ and λ are the sparsity measure (which counts the number of non-zero coefficients) and a constant multiplier respectively. This problem is NP-hard in general. Therefore various relaxed sparsity measures have been presented to make the problem tractable. A commonly used class of measures are $||\mathbf{x}||_p^p = \sum_i |x_i|^p$ with $0 < p \leq 1$.

When the generative model for the signals is unknown, appropriate dictionary learning algorithms can be used to adaptively find better dictionaries for a set of training samples. We are thus searching for a set of elementary functions that allow the set of training signals to be represented sparsely and with a small approximation error.

In this paper we consider the dictionary learning problem as a constrained optimization problem with two sets of parameters, coefficient matrix and dictionary. The constraints are generalizations of those in [1]. The proposed constrained optimization problem is converted into an unconstrained optimization problem using Lagrangian multipliers. We then present reasonably fast methods to update the dictionary. A comparison between the proposed method and other dictionary learning methods is presented.

2. DICTIONARY LEARNING METHODS

In dictionary learning, one often starts with some initial dictionary and finds sparse approximations of the set of training signals whilst keeping the dictionary fixed. This is followed by a second step in which the sparse coefficients are kept fixed and the dictionary is optimized. This algorithm runs for a specific number of alternating optimizations or until a specific approximation error is reached. The proposed method is based on such an alternating optimization (or block-relaxed optimization) method with some advantages over the current methods in the condition and speed of convergence.

If the set of training samples is $\{\mathbf{y}^{(i)} \; : \; 1 \leq i \leq L\}$, where L is the number of training vectors, then sparse approximations are often found (for all $i: \; 1 \leq i \leq L$) by,

$$\min_{\mathbf{x}^{(i)}} \Phi_i(\mathbf{x}^{(i)}) \; ; \; \Phi_i(\mathbf{x}) = ||\mathbf{y}^{(i)} - \mathbf{D}\mathbf{x}||^2 + \lambda ||\mathbf{x}||_p^p \qquad (2)$$

An alternative to minimizing (2) individually on each vector is to find a joint sparse approximation of the matrix $\mathbf{Y} = [\mathbf{y}^{(1)} \; \mathbf{y}^{(2)} \; ... \; \mathbf{y}^{(L)}]$ by employing a sparsity measure in matrix form. The sparse matrix approximation problem can be formulated as,

$$\min_{\mathbf{X}} \Phi(\mathbf{X}) \; ; \; \Phi(\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}), \qquad (3)$$

where $J_{p,q}(\mathbf{X})$ is defined as ([2]),

$$J_{p,q}(\mathbf{X}) = \sum_{i \in I} [\sum_{j \in J} |x_{ij}|^q]^{p/q}. \qquad (4)$$

$||\mathbf{X}||_F = J_{2,2}^{1/2}(\mathbf{X})$ would be the Frobenius-norm. When $p = q$ all elements in $\mathbf{X}$ are treated equally.

The second step in dictionary learning is the optimization of the dictionary based on the current sparse approximation. The cost function in (3) can be thought of as an objective function with two parameters,

$$\Phi(\mathbf{D}, \mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}) \qquad (5)$$

Without additional constraints on the dictionary, minimizing the above objective function is an ill-posed problem. An obvious solution is $\mathbf{D} \to \infty, \mathbf{X} \to 0 \; s.t. \; \mathbf{D}\mathbf{X} = \mathbf{Y}$. By constraining the norm of $\mathbf{D}$ we can exclude these undesired solutions. Dictionaries with fixed column-norms or fixed Frobenius-norm have been used in different papers (for example [3] and [1]). We present the more general admissible sets assuming "'*bounded* column-norm" and "*bounded* Frobenius-norm".

In the Method of Optimal Directions (MOD) [3] the best D is found by using the pseudo inverse of X followed by re-normalizing each atom. The Maximum Likelihood Dictionary Learning algorithm (ML-DL), which is presented in [4], is similar to MOD but uses gradient optimization. If the update is done iteratively, we find the best possible dictionary update without any constraint (similar to MOD). This update is followed by normalizing atoms based on the variance of the corresponding coefficients. The dictionary normalization step in these methods may increase total approximation error. The Maximum *a Posteriori* dictionary learning algorithm (MAP-DL) [1] is based on the assumption that '*a priori*' information is available about the dictionary. By the use of an iterative

This work is funded by EPSRC grant number D000246

method, if the algorithm converges, it finds a dictionary consistent with this *a priori* information [1]. When a fixed column-norm constraint is used, the algorithm updates atom by atom, making the method too slow to be used for many real signals [5].

The K-SVD method presented in [5] is fundamentally different from these methods. In the dictionary update step, the supports of the coefficient vectors (the positions of the non-zero coefficients) is fixed and an update of each atom is found as the best normalized elementary function that matches the errors (calculated after representing the signals with all atoms except the currently selected atom).

The dictionary learning approach proposed in this paper has several similarities with the formulation used in MAP-DL. However, our approach is based on a joint cost function for both, the sparse approximation and the dictionary update and uses a new class of constraints on the desired dictionaries. Furthermore, the algorithms presented to solve the problem are different and are proven to converge. Because the proposed cost functions are not convex, using gradient based methods to update the dictionary will not in general find the global optimum and, like the other methods mentioned above, the algorithms presented in this paper are only guaranteed to find a local minimum.

3. REGULARIZED DICTIONARY LEARNING (RDL)

In this section we consider the dictionary learning problem as a constrained optimization problem.

$$\min_{\mathbf{D},\mathbf{X}} \Phi(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D} \; ; \; \Phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}) \quad (6)$$

where $\mathscr{D}$ is some admissible set. In an iterative two-step optimization scheme, we find the optimum $\mathbf{X}$ with fixed $\mathbf{D}$ in one of the steps. In this paper we use iterative thresholding (IT) [6] for this optimization. In this algorithm a convex function is added to the objective function to decouple the optimization of the x_{ij}. Then the convex function is updated based on the current solution and the algorithm continues with the new objective function. The objective function in (6) and the added convex function have matrix valued parameters leading to a generalization for the IT method.

In every other step of the dictionary learning algorithm we update the dictionary. As noted in [1], two typical constraints are the unit Frobenius-norm and the unit column-norm constraints, both of which lead to non-convex solution sets. In addition to these constraints, the algorithms proposed in this paper can also be used to solve (6) if bounded norm constraints (defined later) are used. With these, the algorithms are guaranteed to find the global optimum within the dictionary update step. Note that (5) is a convex function of **D** (for fixed **X**) and of **X** (for fixed **D**), but it is not convex as a function of both, **X** and **D**, so that the alternating optimization of (3) is not guaranteed to find a global optimum.

Note that if the sparsity measure in the sparse approximation step penalizes coefficients based on their magnitudes (for example $l_p : 0 < p \leq 1$), it is easy to show that the fixed points of the algorithm are on the boundary of the convex sets.

3.1 Constrained Frobenius-Norm Dictionaries

In this section we derive an algorithm for the case in which we constrain the Frobenius-norm of **D**. An advantage of using a constraint on the Frobenius-norm is that the dictionary size can be reduced during dictionary learning by pruning out atoms whose norm becomes small. Another advantage is that the learned dictionary will have atoms with different norms as used in the weighted-pursuit framework [7]. Atoms with large norm then have more chance of appearing in the approximation. It has been shown that the average performance of the sparse approximation increases when the weights are chosen correctly for the class of signals under study [7].

The admissible set for the *bounded* Frobenius-Norm dictionaries is,

$$\mathscr{D} = \{\mathbf{D}_{d\times N} : ||\mathbf{D}||_F \leq c_F^{1/2}\} \quad (7)$$

where c_F is a constant. With the help of a Lagrangian multiplier γ we turn this into an unconstrained optimization problem,

$$\min_{\mathbf{D}} \Phi_\gamma(\mathbf{D},\mathbf{X}), \quad (8)$$

where $\Phi_\gamma(\mathbf{D},\mathbf{X})$ is defined as,

$$\Phi_\gamma(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}) + \gamma(||\mathbf{D}||_F^2 - c_F). \quad (9)$$

The solution to the above minimization problem is a global minimum if the solution satisfies the K.K.T conditions [8, Theorem 28.1]. The admissible set is convex, so any minimum of $\Phi_\gamma(\mathbf{D},\mathbf{X})$ is an optimal solution if $\gamma(||\mathbf{D}||_F^2 - c_F) = 0$. Therefore if $||\mathbf{D}||_F^2 \neq c_F$ then γ must be zero. The objective function is differentiable in **D**. Therefore its minimum is a point with zero gradient. For fixed **X**,

$$\begin{aligned} d\Phi_\gamma(\mathbf{D},\mathbf{X}) &= d\,tr\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X} - \mathbf{X}^T\mathbf{D}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{D}\mathbf{X} \\ &\quad + \mathbf{Y}^T\mathbf{Y}\} + \gamma.\, d\,tr\{\mathbf{D}^T\mathbf{D}\} \\ &= (2\mathbf{X}\mathbf{X}^T\mathbf{D}^T - 2\mathbf{X}\mathbf{Y}^T + 2\gamma\mathbf{D}^T)d\mathbf{D} \end{aligned}$$

$$\Rightarrow \frac{d}{d\mathbf{D}}\Phi_\gamma(\mathbf{D},\mathbf{X}) = 2\mathbf{X}\mathbf{X}^T\mathbf{D}^T - 2\mathbf{X}\mathbf{Y}^T + 2\gamma\mathbf{D}^T = \mathbf{0}$$

$$\Rightarrow \mathbf{D} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I})^{-1} \quad (10)$$

$\Phi_\gamma(\mathbf{D},\mathbf{X})$ is a non-negative convex function of **D** and this solution is minimal. To find the appropriate γ satisfying the K.K.T condition, we note that $\Phi_\gamma(\mathbf{D},\mathbf{X})$ is a continuous function of γ (in the regions in which $(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I})$ is not singular). Therefore if **D** as calculated by (10) and with $\gamma = 0$ is admissible, this **D** is the optimum solution. If (10) does not give an admissible solution, we can use a line-search method to find a $\gamma \neq 0$ such that $||\mathbf{D}||_F = c_F^{1/2}$ (by changing γ in the direction which reduces $|||\mathbf{D}||_F - c_F^{1/2}|$). Interestingly, MOD uses $\mathbf{D} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$, whilst our update uses a *regularized* pseudo inverse.

If we use an equality in the definition of (7) to get the fixed Frobenius-norm constraint, the set becomes non-convex so that we might only find a local minimum, in which case γ could become negative.

3.2 Constrained Column-Norm Dictionaries

The admissible set for the *bounded* column-norm dictionary is defined as,

$$\mathscr{D} = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 \leq c_C^{1/2}\}, \quad (11)$$

where $\mathbf{d}_i$ is the i$^{\text{th}}$ column of the dictionary and c_C is a constant. This admissible set is again a convex set. However, now we need N (number of columns in **D**) Lagrangian multipliers (equal to the number of constraints) and the unconstrained optimization turns to,

$$\min_{\mathbf{D}} \Phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}), \quad (12)$$

where $\Phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X})$ is defined as,

$$\Phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}) + \sum_{i=1}^{N} \gamma_i(\mathbf{d}_i^T\mathbf{d}_i - c_C) \quad (13)$$

With this formulation, the K.K.T conditions are,

$$\forall i : 1 \leq i \leq N, \qquad \gamma_i(\mathbf{d}_i^T\mathbf{d}_i - c_C) = 0\,. \quad (14)$$

This means that for each i if $\mathbf{d}_i^T\mathbf{d}_i$ is not equal to c_C then γ_i should be zero. (12) can be rewritten as

$$\Phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}) + tr\{\mathbf{\Gamma}(\mathbf{D}^T\mathbf{D} - c_C\mathbf{I})\}, \quad (15)$$

where Γ is a diagonal matrix with the γ_is as the diagonal elements. If we use a similar method as before we get an optimum at,

$$\mathbf{D} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \mathbf{\Gamma})^{-1} \tag{16}$$

Even though the minimum seems to be similar to (10), finding $\mathbf{\Gamma}$ is now more difficult as we can no longer use a line search.

Instead of optimizing the original objective function (15) directly we can use an iterative method. By adding a convex function of $\mathbf{D}$ to (15) we get the surrogate function,

$$\Phi^S_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{B},\mathbf{X}) = \Phi_\Gamma(\mathbf{D},\mathbf{X}) + c_s||\mathbf{D}-\mathbf{B}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{B}\mathbf{X}||_F^2 \tag{17}$$

where $\mathbf{B}$ is a $d \times N$ matrix that is set to the previous solution of $\mathbf{D}$ ($\mathbf{D}^{[n-1]}$) in each iteration. c_s is a constant such that $||\mathbf{X}^T\mathbf{X}||_2 < c_s$. To minimize the surrogate function we set the gradient to zero.

$$\frac{d}{d\mathbf{D}}\Phi^S_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{D}^{[n-1]},\mathbf{X}) = -2\mathbf{X}\mathbf{Y}^T + 2\mathbf{X}\mathbf{X}^T\mathbf{D}^{[n-1]T} + 2c_s\mathbf{D}^T - 2c_s\mathbf{D}^{[n-1]T} + 2\mathbf{\Gamma}\mathbf{D}^T = \mathbf{0}$$

$$\Rightarrow \mathbf{D}^{[n]} = (\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_s\mathbf{I} - \mathbf{X}\mathbf{X}^T))(\mathbf{\Gamma} + c_s\mathbf{I})^{-1} \tag{18}$$

All γ_is are non-negative and $(\mathbf{\Gamma} + c_s\mathbf{I})$ is a diagonal matrix. Therefore $(\mathbf{\Gamma} + c_s\mathbf{I})$ is invertible. In equation (18) by changing γ_i we multiply the corresponding column of $\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_s\mathbf{I} - \mathbf{X}\mathbf{X}^T)$ by a scalar and we can *regulate* the norm of each column in $\mathbf{D}$ by the corresponding γ_i. We start with all $\gamma_i = 0$ and for any column of $\mathbf{D}$ for which the norm is more than one, we find the smallest value for γ_i that normalizes that column. In other words, we find $\mathbf{D}_\sharp = \mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_s\mathbf{I} - \mathbf{X}\mathbf{X}^T)$ and then project $\mathbf{D}_\sharp$ onto the admissible set to find $\mathbf{D}^{[n]}$. The algorithm starts with the dictionary $\mathbf{D}^{[0]} = \mathbf{D}_i$ and iteratively reduces the surrogate objective function. We can run the algorithm for a specific number of iterations or stop based on the distance between the dictionaries in two consecutive iterations ($||\mathbf{D}^{[n]} - \mathbf{D}^{[n-1]}||_F < \xi$), for a small positive constant ξ). This iterative method can be shown to converge to the minimum of the original objective function (15) ($\mathbf{X}$ fixed). Alternatively, we can again set the constraint set to have fixed column-norm ($\mathscr{D} = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 = c_c^{1/2}\}$). Here the algorithm may find a local minimum and some of the γ_i might become negative.

4. SIMULATIONS

We evaluate the proposed methods with synthetic and real data. Using synthetic data with random dictionaries helps us to examine the ability of the proposed methods to recover dictionaries exactly (to within an acceptable squared error). We generated the synthetic data and dictionaries as proposed in [1] and [5]. To evaluate the performance on real data, we chose an audio signal, which has been shown to have some sparse structure. We then used the learned dictionary for audio coding and show some improvements in the Rate-Distortion performance.

4.1 Synthetic Data

A 20×40 matrix $\mathbf{D}$ was generated by normalizing a matrix with i.i.d. uniform random entries. The number of non-zero elements was selected between 3 and 7 to generate different sparse coefficient vectors. The locations of the non-zero coefficients were selected uniformly at random. For the unit column-norm dictionary learning, we generated 1280 training samples where the absolute values of the non-zero coefficients were selected uniformly between 0.2 and 1. Iterative Thresholding (IT) [6] was used to optimize (3) using the ℓ_1 measure. This was followed by orthogonal projection onto the selected sub-spaces (to find the best representation in that subspace). The stopping criteria for IT was the distance between two consecutive iterations ($\delta = 3 \times 10^{-4}$) and λ was set to 0.4 . The termination conditions for the iterative dictionary learning methods (RDL and MAP-DL) was set to ($||\mathbf{D}^{[n]} - \mathbf{D}^{[n-1]}||_F < 10^{-7}$).

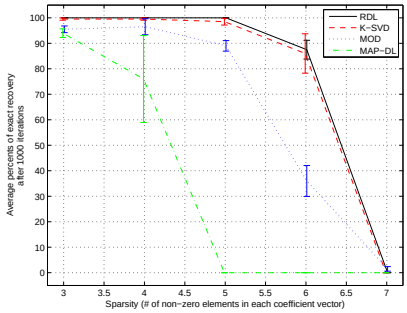

Figure 1: Exact recovery with fixed column-norms dictionary learning.

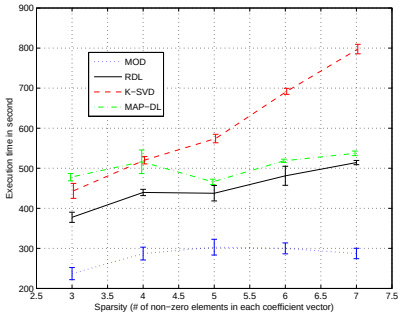

Figure 2: Computation cost of the fixed column-norm dictionary learnings algorithms.

We started from a normalized random $\mathbf{D}$ and used 1000 iterations. The learning parameter (γ) in MAP-DL was selected as described in [1]. We down-scaled γ by a factor of 2^{-j} ($j > 1$) when the algorithm was diverging. To have a fair comparison, we did the simulations for 5 different trials. If the squared error between a learned and true dictionary element was below 0.01, it was classified as correctly identified. The average percentages and standard deviations are shown in Figure 1. It can be seen that in all cases, RDL and K-SVD recovered nearly the same number of atoms and more than the other methods (although for the signals with less than 6 non-zero coefficients, RDL recovered all desired atoms, performance of K-SVD was very close to it). The MAP-DL algorithm did not perform well in this simulation. We guess the reason for this is slow convergence of the approach and the use of more iterations might improve the performance.

In Fig.2 we compare the computation time of the algorithms for the above simulations. Simulations ran on the Intel Xeon 2.66 GHz dual-core processor machines and both cores were used by Matlab. In this graph the total execution time of the algorithms (sparse approximations plus dictionary updates for 1000 iterations) is shown. MOD was fastest followed by our RDL.

We have a larger admissible set when fixing the Frobenius-norm

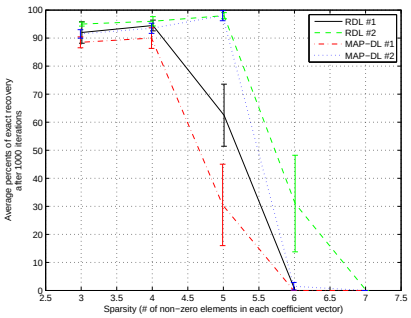

Figure 3: Exact recovery with fixed Frobenius-norm dictionary learning. 1: Desired dictionary had fixed Frobenius-norm. 2: Desired dictionary had fixed column-norms.

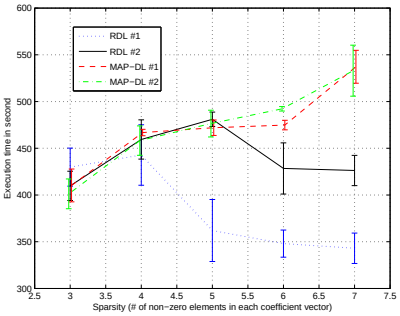

Figure 4: Computation cost of the fixed Frobenius-norm dictionary learning algorithms.

of the dictionary, which makes the problem of exact recovery more complicated and we expect to have less exact recovery for the same sparse signals. For this part we started with normalized random dictionaries, normalized to have either fixed Frobenius-norm or fixed column-norm.

The simulations were repeated for 5 trials and the averages and standard deviations of the atom recovery are shown in the Fig. 3. In these simulations RDL performed slightly better than MAP-DL. The other observation in this figure is that when the desired dictionaries have fixed column-norms, performance of the algorithms increase but do not reach the performance observed when using the more restricted (and appropriate) admissible set. Computation times of the algorithms, on the machines described formerly, are shown in the Fig.4. An interesting observation is the decrease in the computation time of RDL for less sparse signals, when the algorithm could barely recover the correct atoms.

Instead of constraining the dictionaries to have fixed norms, we can use the bounded-norm constraints. To show the possible advantage of these constraints, we repeated the simulations above. The results achieved with these constraints are shown in Fig. 5 We here did the simulations with and without orthogonal projections on the selected spaces found by sparse approximation method. It can be

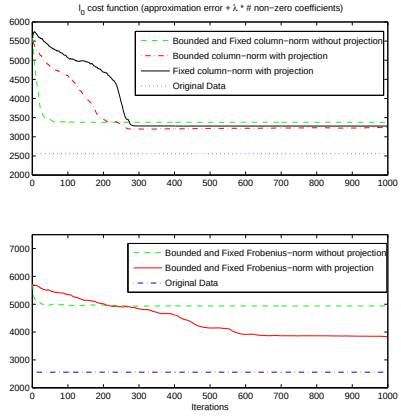

Figure 5: l_0 cost functions of the constrained Frobenius and column -norms dictionary learning algorithms respectively on top and bottom plots.

seen that using bounded-norm admissible set improves performance slightly when constraining the column-norm but it does not change performance of the other method. These plots also show that the orthogonal projection onto the selected spaces can improve overall performances.

4.2 Dictionary Learning for Sparse Audio Representations

In this part we demonstrate the performance of the proposed dictionary learning methods on real data. An audio sample of more than 8 hours was recorded from BBC radio 3, which plays mostly classical music. The audio sample was summed to mono and down-sampled by a factor of 4. From this 12kHz audio signal, we randomly took 4096 blocks of 256 samples each.

In the first experiment we used fixed column-norm and fixed Frobenius-norm dictionary admissible sets. The set of dictionaries with the column-norms equal to c_C is a subset of a larger set of fixed Frobenius-norm dictionaries, when $c_F = Nc_C$. We chose unit column-norm and fixed Frobenius-norm ($c_F = N$) dictionary learning algorithms. We initialized the dictionary with a 2 times overcomplete random dictionary and used 1000 iterations of alternative sparse approximation (using ℓ_1) and dictionary updates. The cost function against iteration, for two different values of λ, are shown in the Fig. 6. This figure shows that the optimal fixed Frobenius-norm dictionaries are better solutions for the objective functions.

As a second experiment we looked at an audio coding example. We used the RDL method with the fixed Frobenius-norm constraint to learn a dictionary based on a training set of 8192 blocks, each 256 samples long. The audio could be modeled using sinusoid, harmonic and transient components. We chose a 2 times overcomplete sinusoid dictionary (frequency oversampled DCT) as the initialization point and ran the simulations with different lambda values for 250 iterations. The number of appearances of each atom ($\lambda = .006$) are sorted based on their ℓ_2 norms and are shown in the Fig. 7. To design an efficient encoder we only used atoms that were used frequently in the representations and therefore shrunk the dictionary. In this test we chose a threshold of 40 (out of 8192) as the selection criteria. This dictionary was used to find the sparse approximations of 4096 different random blocks, each of 256 samples, from the recorded audio. We then coded the location (significant bit map)

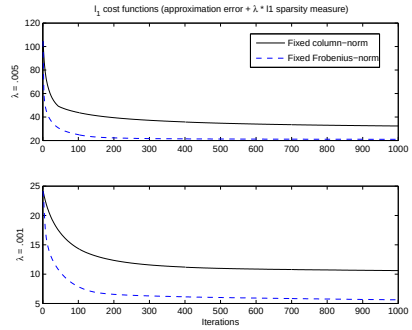

Figure 6: ℓ_1 cost functions for two different Lagrangian multipliers (λ) .005 (top) and .001 (bottom).

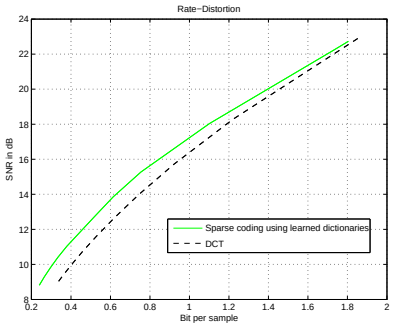

Figure 8: Estimated Rate-Distortion of the audio samples with sparse approximation using learned dictionary and DCT.

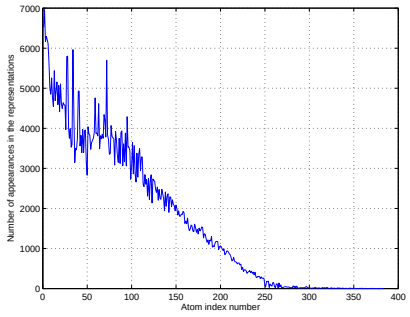

Figure 7: Number of appearances in the representations of the training samples (of size 8192).

and magnitude of the non-zero coefficients separately. In this paper we used a uniform scalar quantizer with a double zero bin. We calculated the entropy of the coefficients to approximate the required coding cost. To encode the significant bit map, we assumed an i.i.d. distribution for the location of the non-zero atoms. The same coding strategy was used to code the DCT coefficients of the same data. The performance is compared in Fig. 8. The convex hull of the rate-distortion performance calculated with different learned dictionaries, each optimized for a different bit-rates, is shown in this figure. Using the learned dictionaries is superior to using the DCT for the range of bit-rates shown, but the advantage is more noticeable for lower rates.

5. CONCLUSIONS

We have formulated the dictionary learning problem as a constrained minimization of a joint cost function. This allowed the derivation of a stable algorithm for dictionary learning, which was shown to perform well on several test data sets. The derived methods differ from most of the previously proposed approaches, such as K-SVD and MAP-DL with unit column-norm *a priori* information, which are based on atom-wise dictionary updates. The proposed methods update the whole dictionary at once. The computation cost of the algorithms were compared and it was found that the proposed methods performed better than, or similar to other competitors. Another simulation showed that using a bounded norm constraint was slightly better or at least as good as a fixed norm constraint. However, more simulations are needed. An alternative to the proposed method, when the constraint set is convex, is iterative gradient projection. This method is similar to the method that was used in Section 3.2 but with a different, and sometimes adaptive, c_s. The overall performance comparison of these methods is the next step of this project.

REFERENCES

[1] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.

[2] S.F. Cotter, B.D. Rao, K. Engan, and K. Kreutz Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[3] K. Engan, S.O. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[4] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[5] M. Aharon, E. Elad, and A.M. Bruckstein, "K-SVD: an algorithm for desiginining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[6] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math*, vol. 57, pp. 1413–1541, 2004.

[7] O. Divorra Escoda, L. Granai, and P. Vandergheynst, "On the use of a priori information for sparse signal approximations," *IEEE Trans. on Signal Processing*, vol. 54, no. 9, pp. 3468–3482, 2006.

[8] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

PARSIMONIOUS DICTIONARY LEARNING

Mehrdad Yaghoobi, Thomas Blumensath, Michael E. Davies

Institute for Digital Communications,
Joint Research Institute for Signal and Image Processing,
The University of Edinburgh, EH9 3JL, UK
{ m.yaghoobi-vaighan, thomas.blumensath, mike.davies}@ed.ac.uk

ABSTRACT

Sparse modeling of signals has recently received a lot of attention. Often, a linear under-determined generative model for the signals of interest is proposed and a sparsity constraint imposed on the representation. When the generative model is not given, choosing an appropriate generative model is important, so that the given class of signals has approximate sparse representations. In this paper we introduce a new scheme for dictionary learning and impose an additional constraint to reduce the dictionary size. Small dictionaries are desired for coding applications and more likely to "work" with sub-optimal algorithms such as Basis Pursuit. Another benefit of small dictionaries is their faster implementation, e.g. a reduced number of multiplication/addition in each matrix vector multiplication, which is the bottleneck in sparse approximation algorithms.

Index Terms— Sparse Approximation, Dictionary Learning, Majorization Method, Sparse Coding

1. INTRODUCTION

Let $\mathscr{Y} = \{y^{(i)} : 1 \leq i \leq L\}$ be a given set of training samples and $\mathscr{X} = \{x^{(i)} : 1 \leq i \leq L\}$ be the corresponding coefficient vectors. $\mathbf{Y}_{d\times L}$ and $\mathbf{X}_{N\times L}$ are the matrices generated by using the elements of $\mathscr{Y}$ and $\mathscr{X}$ as the column vectors, respectively. The dictionary learning problem can be formulated as follows. Given $\mathbf{Y}$, find a "dictionary" matrix $\mathbf{D}$ and a coefficient matrix $\mathbf{X}$, such that the error $\epsilon = \mathbf{Y}-\mathbf{D}\mathbf{X}$ is small and $\mathbf{X}$ is sparse. This is a challenging problem and researchers from different fields have introduced algorithms to solve it approximately [1–4]. Regardless of the sparsity measure, dictionary learning is a non-convex optimization problem and a locally optimum dictionary is often found [5]. Various additional constraints have been recently imposed on the dictionaries to constrain the dictionary search space. These constraints may come from *a priory* information about the dictionary [6,7] or help to attain a fast implementation [8,9].

One application of sparse approximation is sparse coding. In conventional sparse coding, indices of the selected columns of $\mathbf{D}$, called "atom", and the associated coefficients are coded separately [10–12]. The coding cost of specifying the selected atoms is reduced by reducing the size of the dictionary. Therefore minimum size dictionaries are more desirable for a coding purpose. Also, when the size of the learnt dictionary reduces, matrix-vector multiplication can be done faster.

The application of parsimonious dictionary learning is not limited to coding. Dictionary size selection is also a challenging problem in the sparse approximation of real signals. When the size of the dictionary is unknown, one can start with a oversized dictionary and find the minimum size learnt dictionary.

We here introduce a framework for parsimonious dictionary learning. The problem formulation is followed by a practical algorithm to find an approximate solution. We show that the proposed framework gives promising results in dictionary recovery. We then show that the learnt dictionary has advantages over the currently used dictionaries for sparse coding.

This work is funded by EPSRC grant number D000246.

2. PARSIMONIOUS DICTIONARY LEARNING FORMULATION

Dictionary learning can be formulated as the minimization of a joint objective function based on $\mathbf{D}$ and $\mathbf{X}$.

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X}} \phi(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D}; \\ &\phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}), \end{aligned} \tag{1}$$

where $\|.\|_F$ is the Frobenius-norm, $\mathbf{D}$ is a dictionary in an admissible set $\mathscr{D}$ and $J_{p,p}$ is the penalty term over the diversity of the coefficients,

$$J_{p,q}(\mathbf{X}) = \sum_{i\in I}[\sum_{j\in J}|x_{ij}|^q]^{p/q}, \tag{2}$$

where $p \leq 1$. λ is a Lagrangian multiplier. In this paper we use $p = 1$ which makes the minimization over $\mathbf{X}$ convex, if $\mathbf{D}$ is fixed. Various admissible sets have been used for dictionary learning (e.g. see [5]). We use bounded column-norm and bounded Frobenius-norm sets as the admissible sets to make the dictionary update a convex problem for a fixed $\mathbf{X}$. The bounded column-norm admissible set is defined as follows,

$$\mathscr{D}_F = \{\mathbf{D}_{d\times N} : ||\mathbf{D}||_F \leq c_F^{1/2}\}, \tag{3}$$

where c_F is a constant. The bounded Frobenius-norm admissible set is defined by,

$$\mathscr{D}_C = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 \leq c_C^{1/2}\}, \tag{4}$$

where $\mathbf{d}_i$ is the i$^{\text{th}}$ column of the dictionary $\mathbf{D}$ and c_C is a constant. To get a dictionary of minimum size, we now include an additional penalty on the dictionary size. The new joint optimization problem is as follows,

$$\min_{\mathbf{X},\mathbf{D}} \phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D};$$

$$\phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + \theta\| \max_i |\{\mathbf{D}\}_{i,j}\||_0.$$

where $||.||_0$ is an operator that counts the number of non-zero elements, and is therefore related to the size of the dictionary, and $\{\mathbf{D}\}_{i,j}$ is the element (i,j) of $\mathbf{D}$. Because $\phi_{\theta,0,\infty}$ is non-convex and non-continuous, we replace the objective function with a relaxed version as follows,

$$\min_{\mathbf{X},\mathbf{D}} \phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) \; s.t. \; \mathbf{D} \in \mathscr{D};$$

$$\phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + \theta J_{1,q}(\mathbf{D}^T) \quad (5)$$

where $q \geq 1$. By selecting $q = 1$, the objective function penalizes any non-zero element of the dictionary. With some changes, this would be useful for sparse dictionary learning as introduced in [13]. When $q > 1$, the objective function penalizes the number of atoms more than the sparsity of the atoms which is our aim in this paper. The parameter θ is then the regularization parameter which controls the sparsity of the dictionary. By increasing θ, one can get a smaller dictionary.

This objective function can be minimized using alternating minimization. Although this method is guaranteed to reduce the objective in each step, the objective function is not convex and has various local minima. The proposed method optimizes $\mathbf{X}$ and $\mathbf{D}$ alternately keeping the other parameter is fixed. In this framework, the non-convex optimization problem is broken into two convex optimization problems, which can be solved using any convex optimization method. Here we use a majorization minimization method.

3. MAJORIZATION METHOD FOR SPARSE APPROXIMATION AND DICTIONARY UPDATE

We use the majorization minimization method [14] to minimize (5). In the majorization method, the objective function is replaced by a surrogate objective function which majorizes it and can be minimized easier. Here we are interested in the surrogate functions in which the parameters are decoupled, so that the surrogate function can be minimized element-wise.

A function ψ majorizes ϕ when it satisfies the following conditions,

$$\begin{aligned} &\phi(\omega) \leq \psi(\omega,\xi), \; \forall \omega,\xi \in \Upsilon \\ &\phi(\omega) = \psi(\omega,\omega), \; \forall \omega \in \Upsilon, \end{aligned} \quad (6)$$

where Υ is the parameter space. The surrogate function has an additional parameter ξ. We choose this parameter as the current value of ω and find the optimal update for ω.

$$\omega_{new} = \arg\min_{\omega \in \Upsilon} \psi(\omega,\xi). \quad (7)$$

We then update ξ with ω_{new}. The algorithm continues until we find an accumulation point. In practice the algorithm could be terminated when the distance between ω and ω_{new} is less than a threshold.

There are different ways to derive a surrogate function. Jensen's inequality and Taylor series have often been used for this purpose [14]. When $\mathbf{D}$ or $\mathbf{X}$ are fixed, the surrogate function for the quadratic part of (5) can be found [15] by adding $\pi_{\mathbf{X}}(\mathbf{X},\mathbf{X}^{[n-1]}) := c_X||\mathbf{X} - \mathbf{X}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X} - \mathbf{D}\mathbf{X}^{[n-1]}||_F^2$ or $\pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}) := c_D||\mathbf{D} - \mathbf{D}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X} - \mathbf{D}^{[n-1]}\mathbf{X}||_F^2$ respectively, where $c_X > ||\mathbf{D}^T\mathbf{D}||$ and $c_D > ||\mathbf{X}^T\mathbf{X}||$ are two constants and $||.||$ is defined as the spectral norm. $\mathbf{X}^{[n-1]}$ and $\mathbf{D}^{[n-1]}$ are the old values of $\mathbf{X}$ and $\mathbf{D}$ respectively which are the auxiliary parameter ξ in the surrogate objective. In the next two subsections, we show how this method can be used for optimizing (5) in an alternating minimization scheme.

3.1. Matrix Valued Sparse Approximation

In this subsection we briefly show how the majorization method is used for matrix valued sparse approximation. We add $\pi_{\mathbf{X}}$ to (5) and minimize the surrogate objective based on $\mathbf{X}$, followed by updating $\mathbf{X}^{[n-1]}$ with the new value of $\mathbf{X}$. Let $\mathbf{A} := \frac{1}{c_X}(\mathbf{D}^T\mathbf{Y} + (c_X\mathbf{I} - \mathbf{D}^T\mathbf{D})\mathbf{X}^{[n-1]})$. It can be shown that (7) can be solved, for the proposed surrogate objective, by shrinking elements in $\mathbf{A}$, as follows:

$$\{\mathbf{X}^{[n]}\}_{i,j} = \begin{cases} a_{i,j} - \lambda/2 \; sign(a_{i,j}) & \lambda/2 < |a_{i,j}| \\ 0 & otherwise. \end{cases} \quad (8)$$

The convergence of this algorithm is studied in [16] for vector valued coefficients. This proof can also be extended to matrix valued problems.

3.2. Dictionary Update

The objective function is convex when $\mathbf{X}$ is fixed. For fixed $\mathbf{X}$, to minimize over $\mathbf{D}$, the joint sparsity penalty is decoupled by adding $\pi_{\mathbf{D}}$ to the objective function,

$$\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) = \phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}). \quad (9)$$

By separating the terms depending on $\mathbf{D}$, the surrogate cost can be written as,

$$\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) \propto c_s tr\{\mathbf{D}\mathbf{D}^T - 2\mathbf{B}\mathbf{D}^T\} + J_{1,q}(\mathbf{D}^T) \quad (10)$$

where $\mathbf{B} := \frac{1}{c_D}(\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}\mathbf{X}^T))$. The dictionary constraint is introduced into the objective function using Lagrangian multipliers. Let $\mathbf{d}_j$ and $\mathbf{b}_j$ be the j^{th} columns of $\mathbf{D}$ and $\mathbf{B}$ respectively. The objective function, using the bounded column-norm (4), can be written as,

$$\begin{aligned} \psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) &\propto \sum_j (tr\{\tau_j^2\mathbf{d}_j\mathbf{d}_j^T - 2\mathbf{b}_j\mathbf{d}_j^T\} + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q) \\ &= \sum_j (\tau_j^2\mathbf{d}_j^T\mathbf{d}_j - 2\mathbf{d}_j^T\mathbf{b}_j + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q) \\ &\propto \sum_j ((\tau_j\mathbf{d}_j - \mathbf{b}_j/\tau_j)^2 + \tfrac{\theta}{c_D\tau_j}||\tau_j\mathbf{d}_j||_q) \\ &= \sum_j \psi_q^{\frac{\theta}{c_D\tau_j}}(\tau_j\mathbf{d}_j, \mathbf{b}_j/\tau_j) \end{aligned} \quad (11)$$

where $\psi_q^\alpha(\mathbf{v},\mathbf{w}) = (\mathbf{w}-\mathbf{v})^2 + \alpha||\mathbf{v}||_q$, $\tau_j = (1+\gamma_j/c_D)^{1/2}$ and γ_j are the Lagrangian multipliers. To minimize (11), we can minimize the first term by minimizing ψ_q^α for each $\mathbf{d}_j$ independently. With the help of two lemmas presented in [17], we can find the optimum of ψ_q^α based on $\mathbf{d_j}$ for $q = 1, 2$ and ∞. The minimum of $\psi_q^\alpha(\mathbf{v},\mathbf{w})$ based on $\mathbf{v}$ [17, Lemma 4.1] is,

$$\min_{\mathbf{v}} \psi_q^\alpha(\mathbf{v},\mathbf{w}) = \mathbf{w} - \mathscr{P}_\alpha^{q'}(\mathbf{w}) \quad (12)$$

where $\mathscr{P}_\alpha^{q'}$ is the orthogonal projection onto the dual norm ball with radius $\mathbf{w}$ and the dual norm is defined as $||.||_{q'}$ with $1/q' + 1/q = 1$. This minimization problem can be solved analytically for some

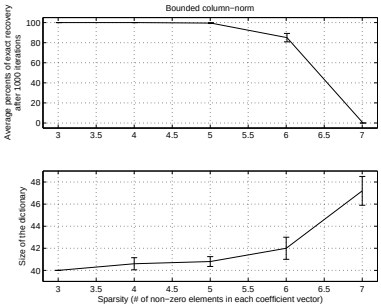

Fig. 1. Exact recovery with the constrained column-norm.

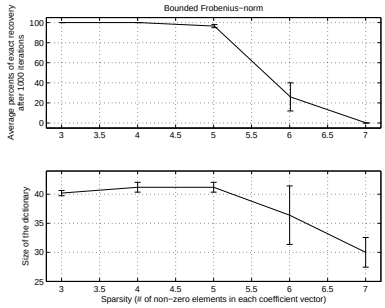

Fig. 2. Exact recovery with the bounded Frobenius column-norm.

q [17, Lemma 4.2]. In this paper we derive the dictionary update formula for $q = 2$.

$$\begin{aligned}\mathbf{b}_j^* &= \arg\min_{\mathbf{d}_j} \psi_2^{\frac{\theta}{c_s \tau_j}}(\tau_j \mathbf{d}_j, \mathbf{b}_j/\tau_j) \\ &= \begin{cases} \frac{1}{\tau_j^2}(1 - \frac{\theta}{2c_D||\mathbf{b}_j||_2})\,\mathbf{b}_j & \frac{\theta}{2c_D} < ||\mathbf{b}_j||_2 \\ 0 & otherwise\,. \end{cases}\end{aligned} \tag{13}$$

When all γ_j are non-negative, for any inadmissible $\mathbf{b}_j^*$ with $\tau_j = 1$ ($\gamma_j = 0$), one can decrease $||\mathbf{d}_j^*||_2$ to $c_c^{1/2}$ by increasing τ_j to satisfy the K.K.T conditions. The dictionary update is therefore done by calculating $\mathbf{B}$ followed first by (13) ($\tau_j = 1$) and secondly by orthogonal projection onto the convex set (4).

When we are looking for a bounded Frobenius-norm dictionary, the dictionary update could be derived using a similar approach, using orthogonal projection onto (3) instead of (4).

4. SIMULATION

We evaluate the proposed method with synthetic and real data. Using synthetic data with random dictionaries helps us to examine the ability of the proposed methods to recover dictionaries exactly (to within an acceptable squared error). To evaluate the performance on real data, we chose audio signals. We then used the learnt dictionary for audio coding and show improvements in Rate-Distortion performance compared to coding with classical dictionaries.

4.1. Synthetic Data

A 20×40 matrix $\mathbf{D}$ was generated by normalizing a matrix with i.i.d. uniform random entries. The number of non-zero elements in each of the coefficient vectors was selected between 3 and 7. The locations of the non-zero coefficients were selected uniformly at random. We generated 1280 training samples where the absolute values of the non-zero coefficients were selected uniformly between 0.2 and 1. We debiased all the sparse approximations by orthogonally projecting onto the space spaned by atoms with non-zero coefficients.

We assume that the desired dictionary size is unknown but bounded. The simulations were started with four times overcompete dictionaries (two times larger than the desired dictionary size). The dictionary updates were based on the joint sparsity objective function (5) (with $\theta = 0.05$, $p = 1$ and $q = 2$). The average percentage of exact atom recovery, i.e. absolute inner product of the learnt atom with one of the atoms in the original dictionary is more than 0.99, for 5 trials are shown in Fig. 1 and 2. We plotted the percentage of the exact recovery of the original atoms, regardless of the learnt dictionary size. In the lower plot, we show the size of dictionary after 1000 iterations. With this θ we identified the size correctly but for less sparse signals (higher k) we got less accurate results.

4.2. Parsimonious Dictionary Learning for Sparse Audio Coding

In this section we demonstrate the performance of the proposed dictionary learning method on audio signals. An audio sample of more than 8 hours was recorded from BBC radio 3, which plays mostly classical music. We used the proposed method with the bounded Frobenius-norm constraint to learn a dictionary based on a training set of 8192 blocks, each 1024 samples long.

In this experiment, instead of fully optimizing over one parameter ($\mathbf{X}$ or $\mathbf{D}$) before switching to the other one, we update each parameter for a small number of iterations and then switch to the other one. This type of alterante optimization was found to be faster in practice.

We chose a 2 times overcomplete sinusoid dictionary (frequency oversampled DCT) as the initialization point and ran the simulations with different lambda values for 5000 iterations of alternative optimization of (11). The number of appearances of each atom, which are sorted based on their ℓ_2 norms, are shown in Fig. 3. To design an efficient encoder we only used atoms that were used frequently in the representations. Therefore we were able to further shrink the dictionary size. In this test we chose a threshold of 40 appearances (out of 8192) as the selection criteria. This dictionary was used to find the sparse approximations of 4096 different random blocks, each of 1024 samples, from the same data set. We then encoded the location (significant bit map) and magnitude of the non-zero coefficients separately. In this paper we used a uniform scalar quantizer with a double zero bin size to code the magnitude. We estimated the entropy of the coefficients to approximate the required coding cost. To encode the significant bit map, we assumed an i.i.d. distribution for the location of the non-zero atoms. The same coding strategy was used to code sparse approximations with a two times frequency over-

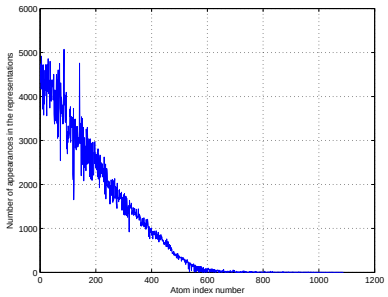

Fig. 3. Number of appearances in the representations of the training blocks (of size 8192).

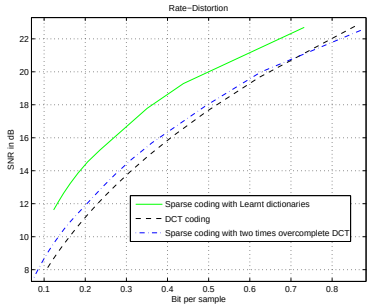

Fig. 4. Estimated Rate-Distortion for the audio coding.

complete DCT (the initial dictionary used for learning) followed by shrinking based on the number of appearances. For reference we calculated the rate-distortation of the DCT coefficient encoding of the same data, using the same method of significant bitmap and non-zero coefficients coding. The performance is compared in Fig. 4. In the sparse coding methods, the convex hulls of the rate-distortion performances calculated with different dictionaries, each optimized and shrunk for different bit-rates, are shown in this figure. Using the learnt dictionaries for sparse approximation is superior to using the DCT or overcomplete DCT for the range of bit-rates shown.

5. CONCLUSIONS

We introduced a formulation for parsimonious dictionary learning. We have shown how we can solve the dictionary learning problem approximately, by imposing a penalty on the size of the dictionary, using a majorization method. A small set of simulations showed that the algorithm often recovers a dictionary with the correct size. We then used the learnt dictionary for sparse coding. We showed the advantages over standard overcomplete and orthogonal dictionaries, specially at low bit-rate. Although the results are promising, more investigations are needed to find a method to determine the parameter θ.

6. REFERENCES

[1] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[2] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.

[3] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.

[4] M. Aharon, E. Elad, and A.M. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[5] M. Yaghoobi, T. Blumensath, and M. Davies, "Regularized dictionary learning for sparse approximation," in *EUSIPCO*, 2008.

[6] T. Blumensath and M.E. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 50–57, 2006.

[7] P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval, "MOTIF: An efficient algorithm for learning translation invariant dictionaries," in *ICASSP*, 2006.

[8] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *ICASSP*, 2005.

[9] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

[10] P. Frossard, P. Vandergheynst, R. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," *IEEE Trans. on Signal Processing*, vol. 52, no. 2, pp. 525–535, 2004.

[11] M.E. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.

[12] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1999.

[13] R. Rubinstein, M. Zibulevsky, and M. Elad, "Sparsity,take 2: Accelerating sparse-coding techniques using sparse dictionaries," submitted.

[14] K. Lange, *Optimization*, Springer-Verlag, 2004.

[15] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," submitted.

[16] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math*, vol. 57, pp. 1413–1541, 2004.

[17] M. Fornasier and H. Rauhut, "Recovery algorithms for vector valued data with joint sparsity constraints," to appear in SIAM Journal of Numerical Analysis, 2007.

Parametric Dictionary Design for Sparse Coding

Mehrdad Yaghoobi[†], and Laurent Daudet[‡], and Mike E. Davies[†],
[†]Institute for Digital Communication, The University of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, UK. yaghoobi@ieee.org, mike.davies@ed.ac.uk
[‡]The Universite Pierre et Marie Curie-Paris VI, Institut Jean le Rond dAlembert-LAM, 11 rue de Lourmel, 75015 Paris, France. daudet@lam.jussieu.fr

***Abstract*—This paper introduces a new dictionary design method for sparse coding of a class of signals. It has been shown that one can sparsely approximate some natural signals using an overcomplete set of parametric functions, e.g. [1], [2]. A problem in using these parametric dictionaries is how to choose the parameters. In practice these parameters have been chosen by an expert or through a set of experiments. In the sparse approximation context, it has been shown that an incoherent dictionary is appropriate for the sparse approximation methods. In this paper we first characterize the dictionary design problem, subject to a minimum coherence constraint. Then we briefly explain that equiangular tight frames have minimum coherence. The parametric dictionary design is then to find an admissible dictionary close to being tight frame. The complexity of the problem does not allow it to be solved exactly. We introduce a practical method to approximately solve it. Some experiments show the advantages one gets by using these dictionaries.**

***Index Terms*—Sparse Approximation, Dictionary Design, Incoherent Dictionary, Parametric Dictionary, Gammatone Filter Banks, Exact Sparse Recovery.**

I. Introduction

SPARSE modeling of signals has recently received much attention as it has shown promising results in different applications. A basic assumption to apply this model is that the given class of signals can be sparsely represented or approximated in an underdetermined linear generative model. In this framework, one can use a matrix $\mathbf{D}_{d\times N} \in \mathbb{R}^{d\times N} : d < N$, called dictionary, to represent the signal approximately using $\mathbf{y} \approx \mathbf{D}\mathbf{x}$. Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the given signal and the coefficient vector respectively. A sparse approximation would be,

$$\widehat{\mathbf{x}} = \arg\min_x \|x\|_0 \ \text{s.t.} \ \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 \leq \xi, \qquad (1)$$

where $\|.\|_0$ is the sparsity measure that counts the number of the non-zero coefficients and ξ is a small positive constant. Because this problem is generally NP-hard, numerous algorithms have been proposed to find an approximate solution. The sparsity of the approximation is increased using an appropriate dictionary for the given class of signals. A dictionary often is selected by concatenating orthogonal bases [3] or using a tight frame [4]. These dictionaries can be improved by dictionary learning methods, see [5] and references therein. These methods adapt an initial dictionary to a set of training samples. Therefore the aim is to *learn* a dictionary for which an input signal, taken from a given class of signals, has a sparse approximation.

This research was fully supported by the UK's EPSRC, grant number D000246/1. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

There is another dictionary selection method, which is called dictionary *design*. Different methods exist to design a suitable $\mathbf{D}$ for a set of natural signals. One method is based on a generative model of the signals. Alternatively, if these signals are to be received by the human sensory system, a more effective method to design $\mathbf{D}$ is to use a human perception model [1], [2]. Here, we assume that the set of elementary functions, which are generated by the proposed model, can be described by using a set of parameters and a parametric function. For example, in the multiscale Gabor functions, the parameters are scale, time and frequency shifts and the parametric function is Gaussian. In general the parameters are in the continuous domain. To generate a dictionary based on these generative functions, we can sample these continuous parameters. The question is then how best to sample the parameters. Several researchers have introduced different methods to optimize the sampling process. In [6], a sampling scheme was introduced which finds an approximately tight frame, using 2D Gabor functions. Alternatively, some researchers optimized the parameters based on the closeness to what is observed in the perceptual systems. In practice, [7] showed that the optimal Gammatone parameters, found by fitting to the human auditory system, do not match the parameters estimated from English speech signals.

When we use an approximate or a relaxed method to find a sparse approximation, having an exact generative model does not guarantee that we find the best sparse approximation. An important parameter of a dictionary, for successful sparse recovery, is its coherence μ [8]. The coherence is defined as the absolute value of the largest inner-product of two distinct atoms and it has been shown that when μ is smaller than a certain threshold MP and BPDN can recover the sparse representation of the input signal [9]. It has also been shown that the coherence upper-bounds the residual error decay in MP [10] and OMP [8]. Therefore a dictionary with small μ is desirable for sparse coding. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of the dictionary. The coherence of $\mathbf{D}$ is the maximum absolute value of the off-diagonal elements of $\mathbf{G}$, whenever the columns of the dictionary are normalized. For such $\mathbf{D}$ if the magnitude of all off-diagonal elements of $\mathbf{G}$ are equal, $\mathbf{D}$ has minimum coherence [11]. This normalized dictionary is called an Equiangular Tight Frame (ETF) [12]. Although this type of frame has various nice properties, we mainly consider its advantages in the exact atom recovery [8] and the residual error decay rate [10]. Unfortunately ETF's do not exist for any arbitrary selection of d and N [12]. Therefore a dictionary design aim can be to find the nearest admissible solution. On the other hand, natural signals do not generally have sparse approximations using an ETF.

2

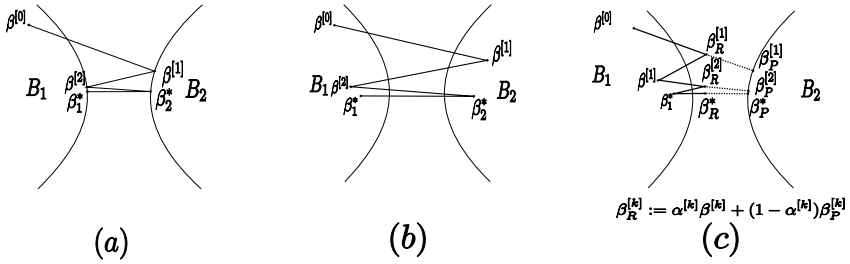

Fig. 1. Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method.

Therefore, the dictionary design problem can be to find a parametric dictionary whose Gram matrix is close to being the Gram matrix of an ETF. This way, domain knowledge is incorporated into the parametric functions and the initial parameters, while the optimization aims at improving the ability of algorithms to find sparse approximations. We expect to have a sparse approximation for the given class of signals using the proposed dictionary. That is because it is generated by sampling the parameters of generative functions fitted to the signal, whilst the dictionary has nice properties that allow exact atom recovery, because it is close to being an ETF. In practice we show that the designed dictionary indeed gives advantages over the standard dictionary, in terms of efficient sparse approximation. Another advantage of the parametric dictionary is that sparse approximation methods only need to store the parameters, instead of the full dictionary, which offers a huge reduction in memory requirement (the size of parameter matrix is much smaller than the size of the corresponding dictionary).

The parametric dictionary design also has some disadvantages. The method is explicitly not a data dependent method. Another difficulty in the given problem is that the current algorithm stores the Gram matrix explicitly. Therefore for a very large block of signal, the current method is not tractable.

A. Contributions of the paper

In this paper we introduce a new framework for dictionary design. To the authors knowledge, this formulation has not been considered previously. This formulation can be used to design a dictionary when dictionary learning is not possible, or is computationally intractable. We show how we can find an approximate solution using an alternating minimization type method.

The parametric dictionary is represented using a small number of parameters (often less than 5). Therefore we do not need to store the dictionary explicitly. This can save a considerable amount of memory when using sparse approximation algorithms.

Finally we show experimentally that there are sparse approximation benefits in using such a parametric dictionary for audio coding.

II. Parametric Dictionary Design: Formulation

In this section we formulate the parametric dictionary design as an optimization problem. Let $\mathbf{D}_\Gamma \in \mathscr{D}$ be a parametric dictionary. Γ is the parameter matrix, with γ_i as its i^{th} column and $\mathscr{D}$ is the set of admissible parametric dictionaries. In this paper, by letting $\mathbf{D}_\Gamma$ be a matrix with the atoms $\mathbf{d}_i$ (with the associated parameters γ_i), we implicitly assume that the generative model is discrete. To select a $\Gamma \in \Upsilon$, where Υ is an admissible parameter set, we can optimize an objective. In section I we explained that for a better performance in sparse coding, we are interested to design a dictionary which is close to being an ETF. For a given normalized $\mathbf{D}$, the coherence of $\mathbf{D}$, $\mu_\mathbf{D}$, is defined by

$$\mu_\mathbf{D} = \max_{i,j:j\neq i} \{|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|\}. \tag{2}$$

A column normalized dictionary $\mathbf{D}_G$ is called ETF, when there is a $\gamma \ : \ 0 < \gamma < \pi/2$.

$$|\langle \mathbf{d}_i, \mathbf{d}_j \rangle| = \cos(\gamma) \ : \ \forall i, j \ i \neq j \tag{3}$$

Strohmer et. al. in [13] showed that if there exists an ETF in $\mathscr{D}$, here the set of d by N uniform frames[1], it is the solution of,

$$\arg\min_{\mathbf{D}\in\mathscr{D}} \{\mu_\mathbf{D}\}. \tag{4}$$

To study the lower bound of μ_D, the existence of an ETF and its Gram matrix, [13] introduced a theorem which shows that when $\mathbf{D} \in \mathbb{R}^{d\times N}$ is a uniform frame, $\mu_\mathbf{D}$ is lower bounded by,

$$\mu_\mathbf{D} \geq \mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{5}$$

Equality holds in (5) *if and only if* $\mathbf{D}$ is an ETF. Furthermore, equality in (5) can only hold if $N \leq \frac{d(d+1)}{2}$.

Let Θ_d^N be the set of Gram matrices of all $d \times N$ ETF's. If $\mathbf{G}_G \in \Theta_d^N$ then the diagonal elements and the absolute values of the off-diagonal elements of $\mathbf{G}_G$ are one and μ_G respectively. A nearness measure of $\mathbf{D} \in \mathbb{R}^{d\times N}$ to the set of ETF's can be defined as the minimum distance between

[1]A frame with unit column norms

Algorithm 1 *Parametric Dictionary Design*

1: **initialization:** $k=1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1\leq i\leq K} : 0<\alpha_i\leq 1$
2: **while** $k \leq K$ **do**
3: $\quad \mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
4: $\quad \mathbf{G}_{P_{k+1}} = \min_{\mathbf{G}\in\Lambda^N} \|\mathbf{G}_{\Gamma_k} - \mathbf{G}\|_F$
5: $\quad \mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$
6: $\quad \mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : \|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}\|_F < \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\}$
7: $\quad k = k+1$
8: **end while**

Algorithm 2 *Parameters Update*

1: **initialization:** $l=1$, $1\leq L$, $\Gamma_k^{[0]} = \Gamma_k$, $\epsilon \in \mathbb{R}^+$, $\phi(\Gamma) = \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}\|_F^2$
2: **for all** $l \leq L$ **do**
3: $\quad \Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$
4: $\quad l = l+1$
5: **end for**
6: $\Gamma_{k+1} = \Gamma_{k+1}^{[L]}$

the Gram matrix of $\mathbf{D}$ and $\mathbf{G}_G \in \Theta_d^N$ [11]. To optimize the distance of a dictionary to an ETF, we can solve,

$$\min_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_G\|_\infty, \tag{6}$$

where the matrix operator $\|.\|_\infty$ is defined as the maximum absolute value of the elements of the matrix. Instead, we would like to use a different norm space which simplifies the problem. An advantage of using ℓ_2 measure in the given problem is that it considers the errors of all elements (and not only the maximum absolute error). In this framework, when there is no ETF in $\mathscr{D}$, we find a dictionary that is close to be quasi-incoherent [8], [10]. Therefore we use the following formulation,

$$\min_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_G\|_F^2, \tag{7}$$

where $\|.\|_F$ is the Frobenius norm. This is a non-convex optimization problem in general. It might have a set of solutions or not have any solution (e.g. Θ_d^N is empty as there do not always exist ETF's for arbitrary N and d). One can extend Θ_d^N to a convex set Λ^N [11], which is non-empty for any N, by

$$\Lambda^N = \{\mathbf{G}\in\mathbb{R}^{N\times N} : \mathbf{G}=\mathbf{G}^T, \operatorname{diag}\mathbf{G} = 1, \max_{i\neq j}|g_{i,j}| \leq \mu_G\}. \tag{8}$$

Relaxing (7), by replacing Θ_d^N with Λ^N, gives the following optimization problem.

$$\min_{\Gamma\in\Upsilon, \mathbf{G}\in\Lambda^N} \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{9}$$

An important difference between (7) and (9) is that the relaxed problem is guaranteed to have at least one solution. We therefore use the relaxed formulation from now on. We show experimentally that the approximate solutions of (9), even though the Gram matrix of the dictionary might only be close to Λ^N, show good performances in sparse approximation.

In the next section we introduce a practical method to find an approximate solution to (9). Our approach has similarities with alternating minimization. This method is guaranteed not to increase the objective function in each step. Because the objective is non-negative, the algorithm is stable due to Lyapunov's second theorem. One can also show that the objective function converges. The stability of the algorithm and the convergence of the objective function do not prove the convergence of the algorithm. The conditions under which the algorithm converges to a set of accumulation points are discussed in Theorem 1. We present a sketch of proof for this theorem and refer the reader to [14] for further details.

III. Parametric Dictionary Design: A Practical Algorithm

A standard method to solve (9) is alternating projection. In this method we alternatingly project the current solution onto the admissible sets, see Fig.1.a. In a finite dimensional setting when the admissible sets are convex, the algorithm converges to a solution in $\mathscr{D}\cap\Lambda^N$ and when $\mathscr{D}\cap\Lambda^N = \emptyset$ to a pair of solutions in $\mathscr{D}$ and Λ^N respectively. In the following, we derive a formulation for the projection onto Λ^N, but there is no easy formulation for the projection onto the set of admissible dictionaries, in general. Therefore we choose a different method which has similarities with alternating minimization, see Fig.1.b. In the alternating minimization framework, we choose the new solutions in $\mathscr{D}$ and Λ^N alternatingly such that the objective does not increase in each update and is thus stable. If the algorithm converges, the fixed point is either in $\mathscr{D}\cap\Lambda^N$, or is a pair of points in $\mathscr{D}$ and Λ^N respectively.

Although the proposed algorithm has similarities with alternating minimization, it does not follow its steps exactly. The difference is that in the stage in which we update the current solution with respect to Λ^N, we choose a point which is somewhere between the current solution and the projection onto Λ^N. Fig.1.c shows a schematic representation of the proposed method. The reason for this modification is that by projection onto Λ^N, the structure of the Gram matrix changes significantly so that the selection of a new point in $\mathscr{D}$ in the following step is very difficult. We can gradually select a closer point to the projected point on Λ^N, when the current $\mathbf{D}_\Gamma$ is close to Λ^N. In the other step, we update $\mathbf{D}$ such that it does not increase the objective in (9).

The parametric dictionary design is summarized in Algorithm 1. In line 4, the algorithm finds the projection onto Λ^N. In line 6, a point in $\mathscr{D}$ is selected which is closer to $\mathbf{G}_{R_{k+1}}$. In the following we show how we calculate the updates in lines 4 and 6.

A. Projection onto Λ^N:

In the objective function (9), $\mathbf{G}$ is a Hermitian matrix. By sign change of any related off-diagonal pair of elements , i.e. $g_{i,j}$ and $g_{j,i}$, we get a new $\widetilde{\mathbf{G}} \in \Lambda^N$. The closest $\mathbf{G}$ to $\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma$, in a Frobenius norm space, is then the $\mathbf{G}$ with a similar sign pattern. For a given $\mathbf{G}_\mathbf{D} = \mathbf{D}^T\mathbf{D} : \mathbf{D}\in\mathbb{R}^{d\times N}$, the projection

of $\mathbf{G}_D$ onto Λ^N can be found by the following operator [11],

$$g_{P i,j} = \begin{cases} \text{sign}(g_{D i,j})\mu_G & i \neq j \\ 1 & otherwise \ , \end{cases} \quad (10)$$

where μ_G is as defined in (5). This operator can be used to find $\mathbf{G}_{P_{k+1}}$ in line 4 of Algorithm 1, by applying into $\mathbf{G}_{\Gamma_k}$.

B. Parameter update:

Let us assume $\mathbf{D}_\Gamma$ is a differentiable function on Υ and therefore (9) is a differentiable function on Υ. An easy way to find Γ_{k+1}, such that it satisfies line 6 of the Algorithm 1, is to use the gradient descent method. We rewrite (9) as a minimization problem based on Γ when $\mathbf{G}_{R_{k+1}}$ is fixed.

$$\min_{\Gamma \in \Upsilon} \phi(\Gamma) \ , \ \phi(\Gamma) := \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}\|_F^2 \quad (11)$$

The gradient of the objective function in (11) can be found by chain rule for the matrix functions [15, D.1.3].

$$\begin{aligned} \nabla_\Gamma \phi &= \nabla_\Gamma \mathbf{D}_\Gamma \ \nabla_{\mathbf{D}_\Gamma} \phi \\ &= 4 \nabla_\Gamma \mathbf{D}_\Gamma \ (\mathbf{D}_\Gamma \mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{D}_\Gamma \mathbf{G}_{R_{k+1}}) \end{aligned} \quad (12)$$

We iteratively use the gradient descent method to find a *local* minimum of the problem (11). Let $\Gamma_k^{[0]} = \Gamma_k$, the updating formula is as follows,

$$\Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi \big|_{\Gamma_k^{[l]}}, \quad (13)$$

where ϵ is a small positive value. In this framework, $\Gamma_{k+1} = \lim_{l \to \infty} \Gamma_{k+1}^{[l]}$. In practice we stop after a given number of iterations or when $\epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$ becomes very small. Algorithm 2 summarizes this parameter update algorithm.

The convergence of Algorithm 1 is guaranteed by the following theorem.

Theorem 1: [14, Theorem 3] Let $\mathbf{D}_\Gamma$ be differentiable. The Algorithm 1 converges to a set of fixed points by starting from $\Gamma_0 \in \Upsilon$, where Υ is a compact set.

We only present a sketch of proof in this paper. We first show that the algorithm reduces the distance of $\mathbf{G}_{\Gamma_k}$ to $\mathbf{G}_k$ in each parameter update. We then show that the objective function of (9) is a continuous function of Γ, which implies the compactness of the solution space. The proof of Theorem 1 is completed by applying Bolzano-Weierstrass theorem, which guarantees existence of at least one accumulation point for the sequences of dictionaries $\{\mathbf{D}_{\Gamma_k}\}_{k \in \mathbb{N}}$. Line 6 of Algorithm 1 prevents the existence of a continuum of accumulation points. Therefore, the accumulation points are fixed points.

IV. Case study

The problem we formulated in this paper is developed in a general form. To show the advantages of using parametric dictionary design in practice, we choose a case study. In sparse audio processing, an important question is how to choose the dictionary [16], [17]. We show that the parametric dictionary design improves the performance of audio sparse approximation and exact recovery based around a Gammatone representation.

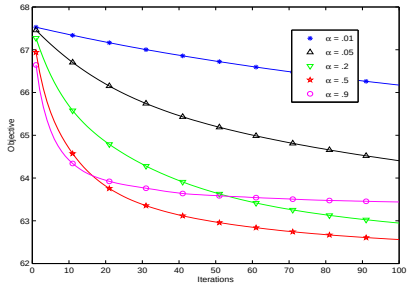

Fig. 2. The objective functions for different $\{\alpha_k\}_{\forall k, \alpha_k = \alpha}$, for a constant α.

A. Gammatone parametric dictionary

The generative function for a Gammatone dictionary is as follows,

$$g(t) = a t^{n-1} e^{-2\pi b B t} \cos(2\pi f_c t) \quad (14)$$

where $B = f_c/Q + b_{min}$, f_c is the center frequency and $n \in \mathbb{N}$, a, b, Q, b_{min} are some constants. The optimal parameter selection is not easy. The dictionary is often generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift. Here, $\gamma = [t_c \ f_c \ n \ b]^T$ are the optimization parameters. The parameters t_c and f_c change the center of the atoms in the time-frequency plane. n and b control the rise time and the width of the atoms in the time domain, respectively. The parameter a is chosen to normalize the atom to unit length. Let $\{\gamma_i\}_{1 \leq i \leq N}$ be a set of the parameters and $g_{\gamma_i}(t)$ be the atom generated using γ_i. The parameter matrix Γ and the parametric dictionary $\mathbf{D}_\Gamma$ are generated using γ_i and $g_{\gamma_i}(\lfloor t f_{samp} \rfloor)$ as the columns respectively, where f_{samp} is the sampling frequency.

To use the gradient discent method for parameter update, $\mathbf{D}_\Gamma$ should be differentiable with respect to Γ. We can extend (14) to a more general function using $n \in \mathbb{R}$. This function is differentiable with respect to Γ. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, Υ is a compact set that guarantees convergence of the algorithm to a set of fixed points. A necessary modification in Algorithm 1 is to use a mapping to Υ, when at least one parameter goes out of Υ, and comparing to the previous solution (to make sure that we do not increase the objective by the parameter update). A simple mapping operator is the thresholding operator, where it chooses the closest admissible parameter.

B. Simulations results

We study the proposed dictionary design method using the Gammatone dictionary discussed in IV. We first investigate the characteristics of the dictionaries throughout the design iterations. We then compare the performance of the initial and the optimized dictionaries in terms of sparse approximation and exact sparse recovery. In all the simulations we choose two times overcomplete dictionaries and window size 1024.

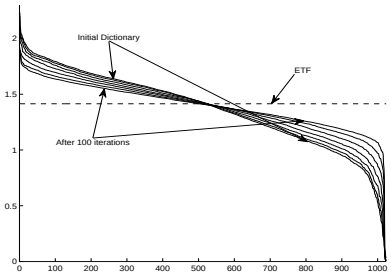

Fig. 3. Eigen values plot of the dictionary.

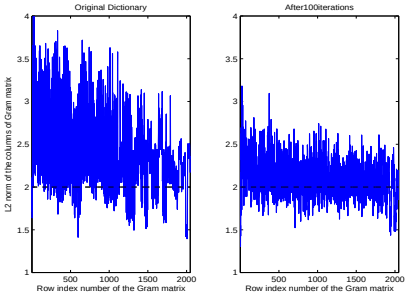

Fig. 4. The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries.

1) Algorithm Evaluation: We evaluate the given algorithm in three different areas. In the first step we show that the algorithm reduces, (or at least keep the same) the objective (9) in each iteration. The parameter B, defined after (14), is the bandwidth of the audio filterbank at the center frequency f_c. We used the fixed values $n = 4$, $Q = 9.26449$, $b_{min} = 24.7$, as they have been suggested in [18], and $b = 0.65$. To generate the initial dictionary, we sampled f_c and t_c. In the method introduced in [19], an extra parameter δ, called step factor, is introduced to indicate the amount of frequency overlap. In this framework the k^{th} frequency center is calculate using the following formula.

$$f_c^k = -Qb_{min} + (f_s/2 + Qb_{min})e^{-k\delta/Q} \qquad (15)$$

f_s is the maximum allowed frequency, which is half of the Nyquist frequency. In our simulations, we choose $\delta = 0.45$. We have chosen a similar method to sample t_c. This time sampling is linear, in contrast with the logarithmic sampling in (15). Let the peak of the envelope of the impulse response of the filter be at t_p and σ indicate the amount of time overlap. The l^{th} time center is found using,

$$t_c^l = t_p + \sigma(l-1)\ t_p\ , \qquad (16)$$

where $\sigma = 0.75$.

To generate a dictionary of $g_{\gamma_i}(t)$, we windowed it to a size equal to the signal length d and made it periodic such that one period is selected as an atom by using the following formula,

$$\mathbf{d}_{\gamma_i,j} = \begin{cases} g_{\gamma_i}(j+d) & 1 \le j < j_{c_i} \\ g_{\gamma_i}(j) & j_{c_i} \le j \le d, \end{cases} \qquad (17)$$

where $j_{c_i} = \lfloor t_{c_i} f_{samp} \rfloor$. We choose a simple sequence of $\{\alpha_k\}$ using $\alpha_k = \alpha$ for all k and a constant α in all simulations. A more complicated sequence might improve the performance of Algorithm 1. However we have not present this here. Instead, we intend to show that the proposed algorithm works in practice, even with a simple $\{\alpha_k\}$. In the first experiment we want to investigate the effect of α. We have plotted the objective function (9) using selected α's, in Fig. 2. As we expect, simulations show reduction of the proposed objectives in each iteration. It is also demonstrated that if α is small, the algorithm converges very slowly. Although using a large α is desireable for a fast convergence, the solution is not as good as the solution found by using a medium range α. For other simulations we use $\alpha = 0.5$ to find a good solution after an acceptable number of iterations.

The proposed algorithm searches for an equiangular *tight frame*. Therefore one way to show the performance of the proposed algorithm is to compare the singular values (SV) of the designed dictionary and the tight frame. A tight frame in $\mathbb{R}^{d\times N}$ has d non-zero SV equal to $\sqrt{N/d}$. We have plotted the sorted SV's of the dictionaries at selected iterations in Fig. 3. It can be seen that the SV's of the designed dictionary get closer to the SV's of the tight frame after each selected number of iterations.

Given that the algorithm is based on distances in the Gram matrix domain, another way to evaluate the algorithm is to show the Gram matrix of the dictionary. We have plotted the ℓ_2 norm of each row of the Gram matrix in Fig. 4. The Gram matrix of the original dictionary and the designed dictionary, after 100 iterations, are shown in the left and right windows respectively. We have shown the ℓ_2 norm of a possible ETF with a dashed line as reference. It can be seen that the Gram matrix of the designed dictionary is closer to the desired Gram matrix.

2) Exact sparse recovery and sparse approximation: In this part we demonstrate the advantages of the parametric dictionary design in terms of exact sparse recovery [8] and sparse approximation. In the first experiment we generate sparse coefficient vectors, with different sparsity, and plot the percentages of the exact recovery for those sparse vectors.

The location of the non-zero coefficients were selected uniformly at random and the PDF of the magnitudes were selected to be Gaussian with zero mean. The Matching Pursuit (MP) algorithm was used to find the sparse approximation. The rate of exact support recovery is calculated by the ratio of the number of correctly found non-zero coefficient places to the number of cases in which at least one location of the zero coefficient was set to be non-zero. We ran the simulations 1000 times. We have shown this ratio as the percentage of

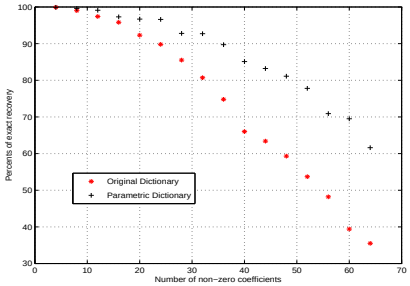

Fig. 5. Exact support recovery of the sparse signals.

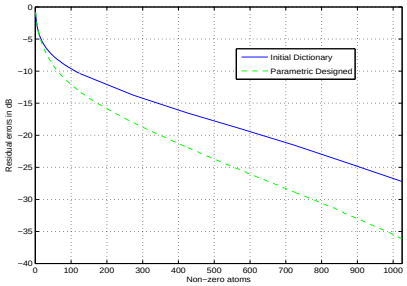

Fig. 6. The residual error using matching pursuit for sparse approximation of the audio signal.

exact recovery in Fig. 5. It is clear that the design method has improved the exact recovery ratio.

For sparse approximation applications, we are more interested to have a dictionary that, if it fails to satisfy exact recovery condition [8], still gives a sparse approximation for a given class of signals. Therefore as the second experiment, we compare the decay rates of the residual error when the MP is used for sparse approximation [10]. We used an audio signal taken from more than 8 hours recorded from BBC Radio 3, which mostly plays classic music. We first down-sampled by a factor of 4 and summed the stereo channels to make a mono signal with 12K samples per second. We used the original Gammatone and the parametric designed dictionaries for 100 blocks, each with the length of 1024 samples. The average decay rate of the residual errors, in logarithmic scales, are shown in Fig. 6. This rate directly influences the performance of sparse approximation methods. That is, we can better approximate the signals with fewer coefficients using a high residual error decay rate dictionary. In Fig. 6, although the curves start with the same slope, after a few iterations, here 10, the designed dictionary shows a clear advantage.

V. Conclusion

We have introduced a signal independent dictionary design method. A parametric function, which is closely related to the given class of signals, was used to design a minimal coherence dictionary. In this framework we have shown that the dictionary design problem is to find an optimal set of parameters. This problem can in general not be solved exactly. Fortunately an approximate solution can be found using the proposed method. In some simulations we showed that A) the given method can find an appropriate set of parameters for the given case study and B) the designed dictionary showed promising performance advantages in terms of exact recovery and sparse approximation. The proposed framework can be extended to include extra constraints, such as to be shift-invariance, quasi-incoherence, data dependence, to have tree structures or structures for fast implementations. That has been left for future work.

References

[1] R. Patterson, M. Allerhand, and C. Giguere, "Time-domain modeling of peripheral auditory processing: a modular architecture and a software platform," *Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1890–1894, 1995.

[2] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.

[3] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, 2003.

[4] E. Candes and L. Demanet, "The curvelet representation of wave propagators is optimally sparse," *Communications on Pure and Applied Mathematics*, vol. 58, no. 11, pp. 1472–1528, 2005.

[5] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," accepted for publication in IEEE Trans. on Signal Processing.

[6] T. Lee, "Image representation using 2d gabor wavelets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[7] S. Strahl and A. Mertins, "Sparse gammatone signal model optimized for english speech does not match the human auditory filters," *Brain Research*, vol. 1220, pp. 224–233, 2008.

[8] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[9] A. Gilbert, S. Muthukrishnan, and M. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. 14th Annu. ACM-SIAM Symp. on Discrete algorithms, Baltimore, MD*, 2003.

[10] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. on Information Theory*, vol. 52, no. 1, pp. 255–261, 2006.

[11] J. Tropp, I. Dhillon, R. Heath Jr., and T. Strohmer, "Designing structural tight frames via an alternating projection method," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.

[12] M. Sustik, J. Tropp, I. Dhillon, and R. Heath, "On the existence of equiangular tight frames," *Linear Algebra and Its Applications*, vol. 426, no. 2-3, pp. 619–635, 2007.

[13] T. Strohmer and R. Heath, "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[14] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," 2009, submitted.

[15] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2005 (v2007.09.17).

[16] M. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.

[17] R. Pichevar, H. Najaf-Zadeh, and L. Thibault, "A biologically-inspired low-bit-rate universal audio coder," in *Audio Engineering Society Convention, Vienna, Austria*, 2007.

[18] M. Slaney, "Lyon's cochlear model," Apple Computer, Tech. Rep., 1988.

[19] ——, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Tech. Rep., 1993.

COMPRESSIBLE DICTIONARY LEARNING FOR FAST SPARSE APPROXIMATIONS

Mehrdad Yaghoobi, Mike E. Davies

Institute for Digital Communications,
Joint Research Institute for Signal and Image Processing,
The University of Edinburgh, EH9 3JL, UK.
{m.yaghoobi-vaighan, mike.davies}@ed.ac.uk

ABSTRACT

By solving a linear inverse problem under a sparsity constraint, one can successfully recover the coefficients, if there exists such a sparse approximation for the proposed class of signals. In this framework the dictionary can be adapted to a given set of signals using dictionary learning methods. The learned dictionary often does not have useful structures for a fast implementation, i.e. fast matrix-vector multiplication. This prevents such a dictionary being used for the real applications or large scale problems. The structure can be induced on the dictionary throughout the learning progress. Examples of such structures are shift-invariance and being multi-scale. These dictionaries can be efficiently implemented using a filter bank. In this paper a well-known structure, called compressibility, is adapted to be used in the dictionary learning problem. As a result, the complexity of the implementation of a compressible dictionary can be reduced by wisely choosing a generative model. By some simulations, it has been shown that the learned dictionary provides sparser approximations, while it does not increase the computational complexity of the algorithms, with respect to the pre-designed fast structured dictionaries.

Index Terms— Sparse Approximation, Dictionary Learning, Compressed Sensing, Compressible Signal, Majorization Minimization.

1. INTRODUCTION

Sparse approximation methods have been successfully applied to various signal processing problems. In this framework we have a linear generative model which can be presented using a full-rank matrix $\mathbf{D} \in \mathbb{R}^{d\times N}$, called dictionary, in the space of discrete signals. Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ respectively be the signal and the coefficient vectors. When $d \leq N$, the generative model is underdetermined and does not have a unique solution. By inducing the sparsity over $\mathbf{x}$ the sparse approximation problem, in a relaxed form, can be formulated as,

$$\arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 + \lambda\mathcal{J}(\mathbf{x}), \tag{1}$$

where $\mathcal{J}(.)$ is the sparsity measure [1], and when it is selected to be the ℓ_1-norm, the objective becomes convex. The convexity of the objective not only helps us to find the global solution of (1), but also guarantees the uniqueness of the solution and, under some conditions, to find the ℓ_0 sparse approximation, where ℓ_0 is the number of non-zero components.

The success of sparse approximation of a given class of signals, is directly determined by choosing a right dictionary, which is often unavailable for the real signals. Various methods have therefore been introduced to select a suitable dictionary. There are two important methods to select a dictionary, which are called dictionary design and dictionary learning, see for example [2–4] and references therein. In this paper we only investigate the dictionary learning problem. A set of training signals $\mathscr{Y} = \{\mathbf{y}_i\}_{i\in\mathcal{I}}$ is given which makes the matrix of training signals $\mathbf{Y} \in \mathbb{R}^{d\times L}$, by putting $\mathbf{y}_i$ as the i^{th} column. The learned dictionary is often found by minimizing an objective based on both $\mathbf{D}$ and $\mathbf{X} \in \mathbb{R}^{N\times L}$ [2, 5, 6], where the latter is the coefficient matrix. In this framework one can find the dictionary by solving the following optimization problem,

$$\arg\min_{\mathbf{D}\in\mathcal{D}} \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\mathcal{J}(\mathbf{X}), \tag{2}$$

where $\mathcal{J}(.)$ is the sparsity measure, which is often column separable, and $\mathcal{D}$ is an admissible set in $\mathbb{R}^{d\times N}$. Different admissible sets have been used to resolve the scale-ambiguity[1] of the optimization problem, e.g. constrained column or Frobenius norms [7].

A new framework is introduced here for the dictionary learning, which its formulation is slightly different to (2), to find a compressible dictionary. The definition of the compressible dictionary is introduced in the next section, followed by some remarks on the features of the compressible dictionaries. In Section 3 a formulation is presented for the Compressible Dictionary Learning (CDL) problem, which is non-convex and difficult to solve exactly. A practical algorithm is then presented in Section 4 to solve the CDL problem *approximately*. By some simulations it has been demonstrated that although the CDL problem is non-convex, the proposed algorithm finds an acceptable sparse dictionary.

2. COMPRESSIBLE DICTIONARY

To impose the compressibility constraint to the dictionary learning problem, we need to introduce the concept of signal compressability [8]. A signal ψ is compressible when the entries obey a power law,

$$|\psi|_{(k)} \leq c_r k^{-r}, \tag{3}$$

where $|\psi|_{(k)}$ is the k^{th} largest value of ψ, $r \geq 1$ and c_r is a constant. In a similar way, we call a matrix $\mathbf{\Psi}$ to be compressible if its entries obey a power law. An important feature of the compressible signals, also has been used in the compressed sensing [9], is that a K-sparse signal approximates a compressible signal with a good approximation. Let $\mathbf{\Psi}_K$ be the matrix of the K largest elements of $\mathbf{\Psi}$, and let the other elements be zero. $\mathbf{\Psi}_K$ is the best estimate for $\mathbf{\Psi}$, in

This work is supported by EPSRC grant number D000246/1.

[1] $\forall\alpha \in \mathbb{R}^+$, $(\mathbf{D}, \mathbf{X})$ and $(\alpha\mathbf{D}, 1/\alpha\mathbf{X})$ have the same approximation errors and the number of non-zero components in the coefficient matrices.

terms of ℓ_2 norm, and the approximation error is upper-bounded by the following formula,

$$\|\mathbf{\Psi} - \mathbf{\Psi}_K\|_F \leq c'_r K^{-r+1/2}. \quad (4)$$

This property has been used in the sensing of a compressible signal by recovering the best K-sparse signal which is a good approximation for the original compressible signal [8].

***Definition* 2.1.** A dictionary $\mathbf{D} \in \mathbb{R}^{d\times N}$ is called *compressible* when for a given full-rank matrix $\mathbf{\Phi} \in \mathbb{R}^{d\times M}$, called the mother dictionary, $\mathbf{D}$ can be generated using the following linear model,

$$\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi}, \quad (5)$$

where $\mathbf{\Psi} \in \mathbb{R}^{M\times N}$ is a compressible matrix and $M \geq d$.

The compressible dictionaries have two important features which are presented by following remarks,

***Remark* 2.1 (Complexity of approximations).** (4) indicates that the approximation error introduced by using $\Psi_{\mathbf{K}}$ is upper bounded. To approximate a compressible dictionary, given $\mathbf{\Phi}$, one can find the best K-sparse $\mathbf{\Psi}_K$. The approximation complexity of $\mathbf{D}$, in general, reduces from $d.N$ to K as a result.

***Proposition* 2.1.** Let $\mathbf{D}$ be a compressible dictionary with the generative model (5) and $|\psi|_{(k)} \leq c_r k^{-r}$. The approximation error of the generated K-sparse dictionary $\mathbf{D}_K = \mathbf{\Phi}\mathbf{\Psi}_K$ decays rapidly by increasing K. The upper-bound of approximation error is as follows,

$$\|\mathbf{D} - \mathbf{D}_K\|_F \leq c'_r \|\mathbf{\Phi}\| K^{-r+1/2},$$

where $\|\mathbf{\Phi}\|$ is the operator norm of $\mathbf{\Phi}$ and c'_r is a constant defined in (4).

One can prove this proposition by using (4) and the definition of operator norm.

***Remark* 2.2 (Fast multiplications).** Any vector multiplication with $\mathbf{D}$ can be done in two steps, a multiplication with the sparse matrix $\mathbf{\Psi}_K$ followed by a multiplication with $\mathbf{\Phi}$. Multiplication with the sparse matrix $\mathbf{\Psi}_K$ is $\mathscr{O}(K)$. When $\mathbf{\Phi}$ has structures which provide fast matrix-vector multiplication, e.g. Fourier and wavelets, the matrix multiplication can be done in $\mathscr{O}(N \log N)$ or better. In the practical applications we are interested in the cases $K \leq N \log N$. Therefore the overall complexity of multiplication with $\mathbf{D}$ is reduced to $N \log N$. It is a significant improvement over the traditional non-structured dictionary multiplication, for example found by dictionary learning, where complexity is $d\ N$.

3. PROBLEM FORMULATION

Let the matrix of training samples $\mathbf{Y} \in \mathbb{R}^{d\times L}$ and the mother dictionary $\mathbf{\Phi} \in \mathbb{R}^{d\times N}$ be given. In the CDL problem, the sparse approximation $\mathbf{X}$ and the dictionary generator matrix $\mathbf{\Psi}$ are unknown. Like the standard dictionary learning problem (2), we can define an appropriate objective function based on $(\mathbf{X}, \mathbf{\Psi})$ and find the dictionary by minimizing the objective. Here we need to add a term to the objective in (2) to promote sparsity of $\mathbf{\Psi}$. Therefore the CDL can be formulated by the following non-convex optimization problem,

$$\begin{aligned} &\arg\min_{\mathbf{\Psi}}\{\min_{\mathbf{X}} \nu(\mathbf{\Psi}, \mathbf{X})\} \ : \\ &\nu(\mathbf{\Psi}, \mathbf{X}) = \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda \mathcal{J}_p(\mathbf{X}) + \gamma \mathcal{J}_q(\mathbf{\Psi}), \end{aligned} \quad (6)$$

where $\mathcal{J}_\rho(\mathbf{\Theta}) = \sum_{i,j} |\theta_{i,j}|^\rho$, for $\rho \in \{p, q\} \leq 1$ and a matrix $\mathbf{\Theta} = \{\theta_{i,j}\}$, is the sparsity measure and $\lambda, \gamma \in \mathbb{R}^+$. Let $p = q = 1$ for simplicity. The sparsity measure $\mathcal{J}_\rho(.)$ is now ℓ_1-norm, which turns (6) into a *bi-convex* optimization problem. The parameters λ and γ control the sparsity of the coefficient and the dictionary generator matrices respectively.

The following lemma shows that (6) is a *well-defined* optimization problem.

***Lemma* 3.1.** *The solution set of the problem (6) is bounded.*

Proof. $\nu(\mathbf{\Psi}, \mathbf{X})$ is a continuous function. Let epigraph of $\nu(\mathbf{\Psi}, \mathbf{X})$ at $(\mathbf{\Psi}^\star, \mathbf{X}^\star)$ be $\mathrm{epi}(\nu, (\mathbf{\Psi}^\star, \mathbf{X}^\star))$. $\mathrm{epi}(\nu, (\mathbf{0}, \mathbf{0}))$ for a continuous function $\nu(\mathbf{\Psi}, \mathbf{X})$ is compact [10]. The solution set is a subset of $\mathrm{epi}(\nu, (\mathbf{0}, \mathbf{0}))$ and therefore bounded. □

The scale ambiguity in the standard dictionary learning is often resolved by constraining $\mathbf{D}$ to be in $\mathcal{D}$. Although the formulation (6) does not have scale ambiguity, it might have non-unique solutions.

***Remark* 3.1.** Let $(\mathbf{\Psi}^*, \mathbf{X}^*)$ be a non-zero solution of (6) and $\alpha := \gamma\mathcal{J}_1(\mathbf{\Psi}^*)/\lambda\mathcal{J}_1(\mathbf{X}^*)$. If $\alpha \neq 1$ then $(\frac{1}{\alpha}\mathbf{\Psi}^*, \alpha\mathbf{X}^*)$ is another solution of (6).

It is worth mentioning the similarity between CDL and the sparse dictionary learning framework [11]. Rubinstein et. al. induced a k-sparsity constraint over each atom and used ℓ_0 as the sparsity measure. In CDL the sparsity is induced over the dictionary, which provides more flexibility in finding sparser dictionary generator matrix. A greedy method has been used in [11] to *approximately* find sparse approximations and dictionary updates. Although no convergence issue has been reported, the mathematical analysis of the algorithm is very difficult. In contrast CDL is guaranteed not only to be stable but also to converge to a set of local minima.

4. CDL ALGORITHM

The problem proposed in Section 3 is non-convex and non-differentiable. The difficulty of the problem can be reduced with the block-relaxation method which has been used for the standard dictionary learning [2]. In this framework, we minimize $\nu(\mathbf{\Psi}, \mathbf{X})$ with respect to $\mathbf{\Psi}$ or $\mathbf{X}$ each time, when the other parameter is kept fixed. In the other words, by starting from an initial solution $(\mathbf{\Psi}^{[0]}, \mathbf{X}^{[0]})$, the algorithm refines the solution by $\mathbf{\Psi}^{[n]} \rightarrow \mathbf{\Psi}^{[n+1]}$ or $\mathbf{X}^{[n]} \rightarrow \mathbf{X}^{[n+1]}$ to reduce $\nu(\mathbf{\Psi}, \mathbf{X})$. When we reduce such a positive objective at each step, the algorithm is stable due to the Lyapunov's second theorem. Due to the continuity of $\nu(\mathbf{\Psi}, \mathbf{X})$, the convergence of the algorithm, to a set of fixed points, can easily be driven using Proposition B.3 of [2].

In the setting introduced in Section 3, $\nu(\mathbf{\Psi}, \mathbf{X})$ is bi-convex and each step of the block-relaxed minimization can be done using a convex optimization method. The majorization minimization method [12] has been chosen to optimize $\nu(\mathbf{\Psi}, \mathbf{X})$ with respect to each parameter. This method is parallelizable and only needs matrix-matrix multiplications, and therefore it is applicable to large scale optimization problems like dictionary learning [2]. A majorizing objective, which is easier to be optimize, is minimized at each step of this method. Recall a function g majorizes f when it satisfies the following conditions,

$$\begin{aligned} f(\omega) &\leq g(\omega, \xi),\ \forall \omega, \xi \in \Upsilon \\ f(\omega) &= g(\omega, \omega),\ \forall \omega \in \Upsilon, \end{aligned}$$

where Υ is the admissible set. The majorizing function has an extra parameter ξ. At each iteration, we first choose this parameter as the current value of ω and find the optimal update for ω.

$$\omega_{new} = \arg\min_{\omega \in \Upsilon} g(\omega, \xi)$$

We then update ξ with ω_{new}. The algorithm continues until we find an accumulation point. In practice the algorithm is terminated when ω and ω_{new} are very close.

The majorizing functions for the $\nu(\mathbf{\Psi}, \mathbf{X})$, when $\mathbf{\Psi}$ or $\mathbf{X}$ is kept fixed, are derived in the next subsections. The majorizing objectives are convex with respect to the corresponding parameters. By letting zero be in the subgradient of the objectives, the update formulas are derived.

4.1. CDL with the majorization method

The objective $\nu(\mathbf{\Psi}, \mathbf{X})$ is an additive combination of the quadratic part $\|\mathbf{\Phi\Psi X} - \mathbf{Y}\|_F^2$, which has bounded curvatures when $\mathbf{\Psi}$ or $\mathbf{X}$ are fixed, and the sparsity measures. A majorizing function can be derived using Taylor series in the matrix form. This operation can simply be done by adding an appropriate strictly convex function to $\nu(\mathbf{\Psi}, \mathbf{X})$, see [2] for more details.

Two distinctive majorizing functions are derived for updating $\mathbf{X}$ and $\mathbf{\Psi}$, for fixed $\mathbf{\Psi}$ and $\mathbf{X}$ respectively. These are followed by deriving the update formulas for each case.

4.1.1. Deriving the update formula for $\mathbf{X}$*:*

Let $\nu_{\mathbf{\Psi}}(\mathbf{X}) : \mathbb{R}^{N\times L} \rightarrow \mathbb{R}^+$ be $\nu(\mathbf{\Psi}, \mathbf{X})$ at a fixed $\mathbf{\Psi}$. The majorizing function is found by adding $\nu_{\mathbf{\Psi}}(\mathbf{X})$ and $\pi_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]})$, which is found by,

$$\pi_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]}) = c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \|\mathbf{X} - \mathbf{X}^{[n]}\|_F^2 - \|\mathbf{\Phi\Psi X} - \mathbf{\Phi\Psi X}^{[n]}\|_F^2,$$

where $c_{\mathbf{\Phi}} > \|\mathbf{\Phi}^T\mathbf{\Phi}\|$ and $c_{\mathbf{\Psi}} > \|\mathbf{\Psi}^T\mathbf{\Psi}\|$. The majorizing objective $\mu_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]})$ is then found by,

$$\mu_{\mathbf{\Psi}}(\mathbf{X}, \mathbf{X}^{[n]}) = tr\{c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi X}^{[n]}) + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]})\} + \lambda \mathcal{J}_1(\mathbf{X}) + const,$$

where $const$ presents the terms which are constant with respect to $\mathbf{X}$. $\mu_{\mathbf{\Psi}}$ is a non-differentiable convex function. The matrix $\mathbf{0}$ is then in the subgradient of $\mu_{\mathbf{\Psi}}$ at the minimum. We know that $\mathbf{X}^{[n+1]} = \arg\min_{\mathbf{X}} \mu(\mathbf{X}, \mathbf{X}^{[n]})$. Therefore $\mathbf{X}^{[n+1]}$ should satisfy,

$$0 \in \partial\mu_{\Psi}(\mathbf{X}^{[n+1]}, \mathbf{X}^{[n]}),$$
$$\partial\mu_{\Psi}(\mathbf{X}, \mathbf{X}^{[n]}) = 2c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X} - 2(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi X}^{[n]}) + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]}) + \lambda \partial\mathcal{J}_1(\mathbf{X}).$$

The update formula for $\mathbf{X}$ can be found by,

$$\mathbf{X}^{[n+1]} = \mathcal{S}_{\lambda/2}\big[\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{\Psi}}}(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi X}^{[n]}) + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]})\big],$$

where $\mathcal{S}_{\lambda/2}$ is the soft-shrinkage operator [13] and $\alpha = \lambda/2$,

$$\mathcal{S}_{\alpha}(\mathbf{A}) = \begin{cases} a_{i,j} - \alpha/2\ sign(a_{i,j}) & \alpha/2 < |a_{i,j}| \\ 0 & otherwise. \end{cases} \quad (7)$$

Algorithm 1 : $\mathcal{CDL}(\mathbf{X}_0, \mathbf{\Psi}_0)$

initialization: $c_{\mathbf{\Phi}} > \|\mathbf{\Phi}^T\mathbf{\Phi}\|$, $K_X, K_\Psi \in \mathbb{N}$
for $t = 0$ **to** T **do**
  $c_{\mathbf{\Psi}} > \|\mathbf{\Psi}^T\mathbf{\Psi}\|$, $\mathbf{X}^{[0]} = \mathbf{X}_t$
  for $n = 0$ **to** $K_X - 1$ **do**
    $\mathbf{X}^{[n+1]} = \mathcal{S}_{\lambda/2}\ [\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{\Psi}}}(\mathbf{\Psi}^T\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi X}^{[n]}) + c_{\mathbf{\Phi}} c_{\mathbf{\Psi}} \mathbf{X}^{[n]})]$
  end for
  $\mathbf{X}_{t+1} = \mathbf{X}^{[K_X]}$
  $c_{\mathbf{X}} > \|\mathbf{X}\mathbf{X}^T\|$, $\mathbf{\Psi}^{[0]} = \mathbf{\Psi}_t$
  for $n = 0$ **to** $K_\Psi - 1$ **do**
    $\mathbf{\Psi}^{[n+1]} = \mathcal{S}_{\gamma/2}\ [\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{X}}}(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi}^{[n]}\mathbf{X})\mathbf{X}^T + c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^{[n]})]$
  end for
  $\mathbf{\Psi}_{t+1} = \mathbf{\Psi}^{[K_\Psi]}$
end for
output: $\mathbf{\Psi}_T$

4.1.2. Deriving the update formula for $\mathbf{\Phi}$*:*

Let $\nu_{\mathbf{X}}(\mathbf{\Psi}) : \mathbb{R}^{M\times N} \rightarrow \mathbb{R}^+$ be $\nu(\mathbf{\Psi}, \mathbf{X})$ at a fixed $\mathbf{X}$. A technique, similar to what was used in 4.1.1, can be used to generate the majorizing function for $\nu_{\mathbf{X}}(\mathbf{\Psi})$. Here, $\pi_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^T)$ is calculated by,

$$\pi_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]}) = c_{\mathbf{\Phi}} c_{\mathbf{X}} \|\mathbf{\Psi} - \mathbf{\Psi}^{[n]}\|_F^2 - \|\mathbf{\Phi\Psi X} - \mathbf{\Phi\Psi}^{[n]}\mathbf{X}\|_F^2.$$

where $c_{\mathbf{X}} > \|\mathbf{X}\mathbf{X}^T\|$. The majorizing objective $\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]})$ is now found to be,

$$\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]}) = tr\{c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^T\mathbf{\Psi} - 2\mathbf{\Psi}^T(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi}^{[n]}\mathbf{X})\mathbf{X}^T + c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^{[n]})\} + \lambda \mathcal{J}_1(\Psi) + const,$$

The matrix $\mathbf{0}$ should be in the subgradient of $\mu_{\mathbf{X}}(\mathbf{\Psi}, \mathbf{\Psi}^{[n]})$ at the minimum, $\mathbf{\Psi}^{[n+1]}$. This provides the following update formula,

$$\mathbf{\Psi}^{[n+1]} = \mathcal{S}_{\gamma/2}\big[\frac{1}{c_{\mathbf{\Phi}} c_{\mathbf{X}}}(\mathbf{\Phi}^T(\mathbf{Y} - \mathbf{\Phi\Psi}^{[n]}\mathbf{X})\mathbf{X}^T + c_{\mathbf{\Phi}} c_{\mathbf{X}} \mathbf{\Psi}^{[n]})\big],$$

where $\mathcal{S}_{\gamma/2}$ is again the soft-shrinkage operator (7), with $\alpha = \gamma/2$. Algorithm 1 presents a pseudocode for the CDL method. In this pseudocode, the outer loop switches between the optimizing parameters. The inner loops are for updating each parameter, for a given number of iterations, before switching to the other parameter. It is also possible to choose different methods for switching between optimizing parameters.

5. SIMULATIONS

The CDL has been used to learn a dictionary for sparse audio coding in this section. Table 1 shows the parameters have been used in this simulation. The training matrix $\mathbf{Y}$ was generated by randomly selecting blocks of an audio signal recorded from BBC Radio 3, which often plays classical music. The parameters c_{Φ}, c_{Ψ} and c_X are chosen to be larger than, but close to, the corresponding operator norms to speed up convergence of the CDL.

The generation of a selected atom in the learned dictionary $\mathbf{D}$ is schematically demonstrated in Figure 1. ψ_i is plotted in part (a). This sparse vector, by multiplying to $\mathbf{\Phi}$, generates atom $\mathbf{d}_i$. Therefore the atoms of $\mathbf{\Phi}$, which are related to the non-zero coefficients

Table 1. The parameters of CDL for the sparse audio coding.

d	M = N	L	λ	γ	T	$\mathbf{\Psi}_0$	$\mathbf{X}_0$
256	512	8192	0.02	0.01	1000	$\mathcal{N}(0,1)$	$\mathbf{0}$

of ψ_i, contribute to generate $\mathbf{d}_i$. The plots (b), (c) and (d) demonstrate, respectively, the contributing atoms of $\mathbf{\Phi}$, scaled version of these atoms and $\mathbf{d}_i$.

Now that we have found the learned dictionary, we can show its advantages in the sparse approximation of the audio signals. We chose 4096 different random blocks of samples from the same audio sample. The iterative thresholding method has been used for sparse matrix approximation, using $\lambda = 0.02$. An extra step of CDL is re-normalizing the learned dictionary to the initial Frobenius-norm, to make further comparisons fair. Figure 2 shows the phase-plot of the algorithm. In this phase-plot the horizontal and vertical axes are $\mathcal{J}_1(\mathbf{X})$ and approximation error, respectively. The result shows that for an approximation error, the approximation by using learned dictionary is sparser (has less ℓ_1).

6. CONCLUSION

A novel dictionary model was introduced. Dictionaries that obey this model, called compressible dictionaries, appear to be more suitable for implementation. An optimization problem is then formulated to find a compressible dictionary. This optimization objective is non-convex and non-differentiable. A practical algorithm was introduced to find an approximate solution (local minimum). A compressible dictionary was learned for the audio signals. It was shown that the sparser approximations of the evaluation samples are in average yielded, using the learned dictionary. Further investigations on the recoverability, the convergence proof and the parameter selection have been left for a future work.

7. REFERENCES

[1] K. Kreutz-Delgado, B. D. Rao, K. Engan, T.W. Lee, and T. J. Sejnowski, "Convex/schur-convex (csc) log-priors and sparse coding," in *Joint Symposium on Neural Computation*, 1999.

[2] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," to appear in IEEE Trans. on Signal Processing, 2009.

[3] M. Aharon, E. Elad, and A.M. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[4] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," submitted, 2009.

[5] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.

[6] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.

[7] M. Yaghoobi, T. Blumensath, and M. Davies, "Regularized dictionary learning for sparse approximation," in *EUSIPCO*, 2008.

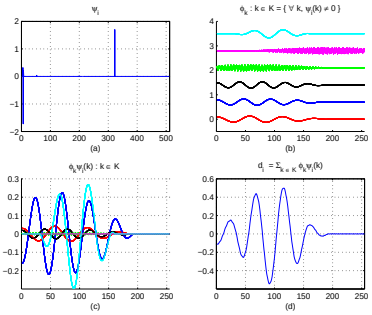

Fig. 1. The atom generation in the CDL framework: (a) i^{th} column of $\mathbf{\Psi}$, ψ_i, (b) The atoms ϕ_k's which are related to the non-zero values of selected ψ_i, $\{\phi_k : \psi_i(k) \neq 0\}$, (c) $\phi_k \psi_i(k) : \psi_i(k) \neq 0$, (d) The i^{th} atom of $\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi}$.

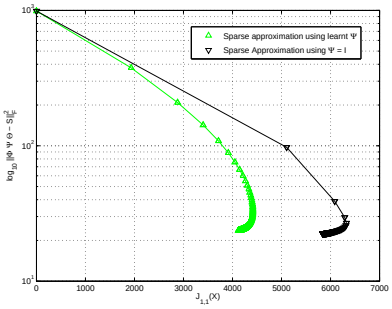

Fig. 2. The phase plots (representation error v.s. ℓ_1) of 4096 evaluation signals.

[8] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[9] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[11] R. Rubinstein, M. Zibulevsky, and M. Elad, "Learning sparse dictionaries for sparse signal representation," submitted, 2008.

[12] K Lange, *Optimization*, Springer-Verlag, 2004.

[13] D.L. Donoho and J.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

STRUCTURED AND INCOHERENT PARAMETRIC DICTIONARY DESIGN

Mehrdad Yaghoobi[†], *Laurent Daudet*[‡], *Michael E. Davies*[†]

[†] Institute for Digital Communications (IDCom), The University of Edinburgh, EH9 3JL, UK

[‡] Institut Langevin (LOA), UMR 7587, Université Paris Diderot - Paris 7, ESPCI, 10 rue Vauquelin 75231 Paris Cedex 05, France.

{ m.yaghoobi-vaighan, mike.davies}@ed.ac.uk, laurent.daudet@espci.fr

ABSTRACT

A new dictionary selection approach for sparse coding, called parametric dictionary design, has recently been introduced. The aim is to choose a dictionary from a class of admissible dictionaries which can be presented parametrically. The designed dictionary satisfies a constraint, here the incoherence property, which can help conventional sparse coding methods to find sparser solutions in average. In this paper, an extra constraint will be applied on the parametric dictionaries to find a structured dictionary. Various structures can be imposed on dictionaries to promote a correlation between the atoms. We choose a useful structure which lets us to implement the dictionary using a set of filter banks. This indeed helps to implement the dictionary-signal multiplications more efficiently. The price we pay for the extra structure is that the designed dictionary is not as incoherent as unstructured parametric designed dictionaries.

Index Terms— Sparse Approximation, Dictionary Selection, Parametric Dictionary Design, Structured Dictionary.

1. INTRODUCTION

Solving an underdetermined linear system inducing a sparsity constraint on the representation has found various applications recently. Often it is assumed that the generative model is known *a priori*. The generative model is often represented by a matrix, called a *dictionary*, $\mathbf{D}_{d\times N} \in \mathbb{C}^{d\times N} : d < N$, which can be used to generate the given signal $\mathbf{y}$ by $\mathbf{y} \approx \mathbf{D}\mathbf{x}$. Each column of $\mathbf{D}$ is called an *atom*. Here we *only* consider real atoms and signals. The sparse approximation would be,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \leq \xi, \qquad (1)$$

where the operator $\|.\|_0$ counts the number of non-zero coefficients and $\xi \in \mathbb{R}^+$ is a small constant. Optimization of (1) is very difficult in general and we often use some kind of relaxations or approximations to make it tractable, see [1] for a survey on different sparse coding methods.

When the dictionary is unknown, it can be adapted to a set of training samples using dictionary learning methods, see for example [2,3]. Alternatively one can generate a dictionary which satisfies some mathematical properties to facilitate the use of dictionary with conventional sparse coding algorithms. Parametric Dictionary Design (PDD) [4] is proposed in such a framework, in which the dictionary is specified by a set of parameters. The aim is to find a set of parameters subject to the incoherence of dictionary. The (mutual) coherence [5] $\mu_{\mathbf{D}}$ of a column normalized dictionary $\mathbf{D}$ is defined as follows,

$$\mu_{\mathbf{D}} = \max_{i,j:j\neq i} \{|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|\}.$$

A dictionary is incoherent if its coherence is small and the largest inner-product of two distinct atoms is thus small. The greedy pursuit and basis pursuit algorithms are more successful in Perfect, or Exact, Recovery [5] and the representations are often sparser using incoherent dictionaries, which is a motivation for the incoherent PDD. By letting the dictionary lie in the parametric space we promote the availability of sparse approximations and by minimizing the coherence of the dictionary we improve the performance of practical sparse coding algorithms.

A drawback of the PDD is that the designed dictionary does not have a useful structure, for example, to enable fast implementation. A structured dictionary is in general a dictionary in which the atoms are correlated. A simple example of structured dictionaries is a shift-invariant dictionary in which the atoms are time-shifted versions of a set of mother atoms. A parametric dictionary is called "structured", if there exist at least two distinct atoms that depend on the values of a single non-empty set of parameters. In this setting, the dictionary is not column separable based on the parameters (the value of a single parameter can change more than one atom). The number of parameters is also reduced as a result, which can help to free up some memory in practice. Although the PDD framework in [4] includes structured dictionaries, they will here be considered with more detail. A case study will be presented later to practically demonstrate the advantages of the proposed method. A new approach for the PDD is also presented which can be used in structured and non-structured scenarios. It simplifies the parameter update step by reducing the problem order from quartic to quadratic form.

The contributions of current paper are twofold:

1. *Presenting a new practical algorithm for solving the parameter update step of PDD:* In the previous reports [4, 6], we introduced a gradient descent based algorithm for the parameter update. Although it works well in some applications, by constraining the search space to the space of rank-d matrices, the parameter update step would be easier. A technique to project onto such a space followed by updating the parameters will later be explored in this paper.

2. *Applying a structure to the parametric dictionaries to accelerate dictionary implementations:* A shift-resilience structure is proposed here. The modified PDD, which is called Structured PDD, is presented and the designed dictionary is compared with the initial and the unstructured dictionary by some simulations.

This work is supported by EU FP7, FET-Open grant number 225913. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

Algorithm 1 *Parametric Dictionary Design*

1: **initialization:** $k = 1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1 \leq i \leq K} : 0 < \alpha_i \leq 1$
2: **while** $k \leq K$ **do**
3: $\quad \mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
4: $\quad \mathbf{G}_{P_{k+1}} = \arg\min_{\mathbf{G} \in \Lambda^N} \|\mathbf{G}_{\Gamma_k} - \mathbf{G}\|_F$
5: $\quad \mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1 - \alpha_k)\mathbf{G}_{\Gamma_k}$
6: $\quad \mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : \|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}\|_F < \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\}$
7: $\quad k = k + 1$
8: **end while**

Algorithm 2 *Parameter Update Step*

1: $\mathbf{G} = \mathbf{G}_{R_{k+1}}$
2: $\mathbf{G}^{\frac{1}{2}} = \Sigma_d^{\frac{1}{2}} \mathbf{U} \; : \; \mathbf{G} = \mathbf{U}\Sigma\mathbf{U}^T$
3: $\mathbf{A}^* = \mathbf{V}\mathbf{W}^T \; : \; \mathbf{D}_{\Gamma_k}\mathbf{G}^{\frac{T}{2}} = \mathbf{V}\Delta\mathbf{W}^T$
4: $\mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathscr{D}} \|\mathbf{D} - \mathbf{A}^*\mathbf{G}^{\frac{1}{2}}\|_F$
5: $\mathbf{D}_{\Gamma_{k+1}} = \begin{cases} \mathbf{D}^* \\ \mathbf{D}_{\Gamma_k} \end{cases}$ see (7) for the criteria.
6: Updating Γ_{k+1} with the parameters of $\mathbf{D}_{\Gamma_{k+1}}$

2. PARAMETRIC DICTIONARY DESIGN

Let $\mathbf{D}_\Gamma \in \mathcal{D}$ be a column normalized[1] parametric dictionary where $\Gamma \in \Upsilon$ is a collection of parameters and $\mathcal{D}$ is an admissible set. In a simple setting, Γ is a matrix in $\mathbb{R}^{p \times N}$ and each atom $\mathbf{d}_i$ can be generated using a column of parameter matrix γ_i. The aim of PDD is to find Γ^* such that the designed dictionary $\mathbf{D}_{\Gamma^*}$ is *incoherent, i.e.* μ is small. The inner-product of two atoms of $\mathbf{D}$ represents the angle between those atoms. A dictionary with uniform angles between each pair of distinct atoms is called an Equiangular Tight Frame (ETF), which has the minimum coherence [7]. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of $\mathbf{D}$. The Gram matrix $\mathbf{G}_G$ of an ETF has unit values on the main diagonal and the absolute values of the off-diagonal elements are μ_G, which is defined as,

$$\mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{2}$$

Let the linear space of full rank matrices in $\mathbb{R}^{d \times N}$ be equipped with the trace inner product, *i.e.* $\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times N} \; \langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}\{\mathbf{A}^T\mathbf{B}\}$. The PDD can be defined as finding a dictionary with a Gram matrix close to the set of Gram matrices of ETF's, Θ_d^N. An ETF *can* exist if $N \leq \frac{d(d+1)}{2}$, which it means that there is no ETF for some pairs of (d, N)'s. To simplify the problem and resolve the issue of empty Θ_d^N's for some (d, N)'s, Θ_d^N is replaced by a convex set Λ^N [8], which includes Θ_d^N, as follows,

$$\Lambda^N = \{\mathbf{G} \in \mathbb{R}^{N \times N} : \mathbf{G} = \mathbf{G}^T, \mathrm{diag}\, \mathbf{G} = 1, \max_{i \neq j} |g_{i,j}| \leq \mu_G\}.$$

The PDD problem can now be reformulated as an optimization problem,

$$\inf_{\Gamma \in \Upsilon, \mathbf{G} \in \Lambda^N} \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{3}$$

In this paper we assume that Υ is a compact set, which lets us to use the "min" operator instead of "inf" in (3). Solving (3) is not easy in general. To simplify the problem and find an approximate solution, we assume that $\mathbf{D}_\Gamma$ is continuously differentiable, *i.e.* class C^1, then apply a relaxed version of the alternating minimization method. In the alternating minimization method, Γ and $\mathbf{G}$ are updated alternatingly to reduce the objective of (3), while the other parameter is kept fixed. The stability, *i.e.* boundedness, of the algorithm is thus guaranteed. A relaxed version of such method has been used in [4] in which Γ is updated to reduce the distance of the Gram matrix and a point between current Gram matrix and the current $\mathbf{G} \in \Lambda^N$. The relaxation is controlled by a scalar parameter α. This point might be outside of both Υ and Λ^N. A pseudocode for this algorithm is presented in Algorithm 1. It has two important steps, line 4 and 6. $\mathbf{G}$ is updated in line 4 with the closest point in Λ^N to the current $\mathbf{G}_{\Gamma_k}$.

[1] In this paper we assume that the dictionary is always column normalized.

As long as Λ^N is convex, $\mathbf{G}_{P_{k+1}}$ is unique and it can be found by projecting $\mathbf{G}_{\Gamma_k}$ onto Λ^N using the following operator [8],

$$g_{P_{i,j}} = \begin{cases} \mathrm{sign}(g_{D_{i,j}})\mu_G & i \neq j \\ 1 & o.w. \end{cases}, \tag{4}$$

where $g_{D_{i,j}}$ is the $(i,j)^{th}$ component of $\mathbf{G}_{\Gamma_k}$. The parameter update step of line 6 can be done using a gradient descent method as introduced in [6]. A difficulty is that the gradient is a tensor and applying conventional optimization methods become difficult in this setting. Here we introduce an alternative technique to update parameters.

Let the set of symmetric rank-d matrices in $\mathbb{R}^{N \times N}$ be noted by $\mathcal{S}^+(d, N)$ [9], which is shown to be equivalent to the set of Gram matrices of full-rank matrices in $\mathbb{R}^{d \times N}$ [10, Proposition 1.1]. $\mathcal{S}^+(d, N)$ has some interesting features which might be useful for the PDD and we left it for an individual research in the future. The first step of the parameter update step can be to find the orthogonal projection of $\mathbf{G}_{R_{k+1}}$ onto $\mathcal{S}^+(d, N)$. If $\mathbf{G}_{R_{k+1}} = \mathbf{U}\Sigma\mathbf{U}^T$, the projection onto $\mathcal{S}^+(d, N)$ can be found by $\mathcal{P}_{\mathcal{S}^+(d,N)}\{\mathbf{G}_{R_{k+1}}\} = \mathbf{U}\Sigma_d\mathbf{U}^T$ where $\Sigma_d = \mathrm{diag}\{\sigma_i\}_{i \in \mathcal{I}_d}$ and $\mathcal{I}_d$ is the set of d largest eigenvalues of $\mathbf{G}_{R_{k+1}}$ [8]. We can now restrict the search space to $\mathcal{S}^+(d, N)$ and find an update which is closer to $\mathcal{P}_{\mathcal{S}^+(d,N)}\{\mathbf{G}_{R_{k+1}}\}$.

$\mathcal{S}^+(d, N) = \{\mathbf{D}^T\mathbf{D} : \mathbf{D}^T \in \mathbb{R}_*^{N \times d}\}$ where $\mathbb{R}_*^{N \times d}$ is the set of all full-rank real $N \times d$ matrices [10]. A further simplification can be to use a mapping from $\mathcal{S}^+(d, N)$ to $\mathbb{R}_*^{N \times d}$ and use a new metric, *i.e.* $\|.\|_F$ in $\mathbb{R}_*^{N \times d}$. This mapping is not unique which is caused by the fact that the Gram matrix is invariant to the left rotation of $\mathbf{D}$. This mapping can be found in two steps, first by calculating $\mathbf{G}^{\frac{1}{2}}$, for example, using eigenvalue decomposition of $\mathbf{G} = \mathbf{U}\Sigma_d\mathbf{U}^T$, *i.e.* $\mathbf{G}^{\frac{1}{2}} = \Sigma_{d\ d \times N}^{\frac{1}{2}}\mathbf{U}^T$, where $\Sigma_d^{\frac{1}{2}} = \mathrm{diag}\{\sigma_i^{1/2}\}_{i \in \mathcal{I}}$ is $d \times N$ diagonal matrix. Then finding the best rotation by minimizing the following objective,

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathbf{R}_*^{d \times d}, \mathbf{A}^T\mathbf{A} = \mathbf{I}_d} \|\mathbf{D}_{\Gamma_k} - \mathbf{A}\mathbf{G}^{\frac{1}{2}}\|_F. \tag{5}$$

This is a standard optimization problem which can be solved exactly [11, Example 7.4.8] as $\mathbf{A}^* = \mathbf{V}\mathbf{W}^T$, where $\mathbf{D}_{\Gamma_k}\mathbf{G}^{\frac{T}{2}} = \mathbf{V}\Delta\mathbf{W}^T$ is a singular value decomposition. Using these two steps we can find a mapping $f : \mathcal{S}^+(d, N) \to \mathbb{R}_*^{d \times N}$ by $f(\mathbf{G}) = \mathbf{A}^*\mathbf{G}^{\frac{1}{2}}$. Let $d(\mathbf{D}, \mathbf{G}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{G}\|_F$. The parameter update can now be found as follows,

$$\mathcal{D}^* = \{\forall \mathbf{D}^* : \mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathscr{D}} \|\mathbf{D} - f(\mathbf{G})\|_F\}, \tag{6}$$

$$\mathbf{D}_{\Gamma_{k+1}} \begin{cases} \in \mathcal{D}^* & d(\mathbf{D}^*, \mathbf{G}_{R_{k+1}}) < d(\mathbf{D}_{\Gamma_k}, \mathbf{G}_{R_{k+1}}) \\ = \mathbf{D}_{\Gamma_k} & o.w. \end{cases} \tag{7}$$

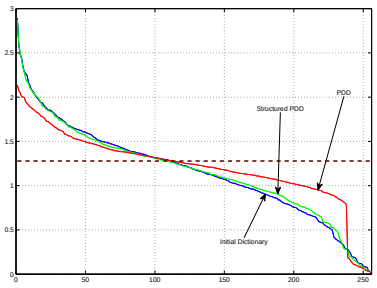

Fig. 1. Eigenvalues of the parametric dictionaries.

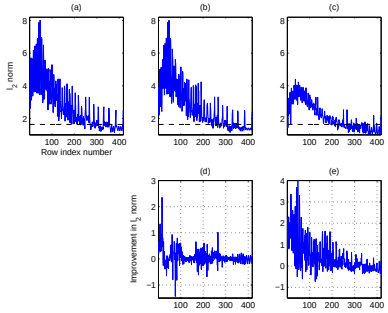

Fig. 2. ℓ_2 norms of the initial (a), the structured designed (b) and unstructured designed dictionaries (c). The improvement of the ℓ_2 norms w.r.t an ETF for the structured designed (d) and unstructured designed (e) dictionaries.

where $\mathscr{D}$ is the set of parametric dictionaries. Note that the solution of (6) might not be unique. In this case we can update with one of the solutions. The reason that we use (7) instead of directly updating $\mathbf{D}_{\Gamma_{k+1}}$ with a $\mathbf{D}^* \in \mathcal{D}^*$ is to prevent a continuum of solutions. There is a wide range of methods to approximately minimize (6), *e.g.* gradient descent, Newton's and Gauss-Newton's methods. The dictionary update step also provides a parameter update which is used in the PDD. A pseudocode for the new parameter update step in line 6 is presented in Algorithm 2.

2.1. Structured Parametric Dictionary Design

A parametric dictionary is called structured if a single parameter affects more than one atom. This framework is general and we only consider a special case, in which the dictionary is partitioned into disjoint sets of uncorrelated atoms. In other words, changing a single parameter can only change the atoms of a partition. An example of such dictionaries will be presented in the next section. Such a dictionary can be presented as $\mathbf{D}_\Gamma = [\mathbf{D}_{\gamma_k}]_{k\in\mathcal{K}}$, where the operator $[.]_{k\in\mathcal{K}}$ is the concatenation of operands. A step in most optimization techniques, which is used for the line 4, is to calculate the gradient of $\mathbf{D}_\Gamma$ with respect to $\{\gamma_k\}_{k\in\mathcal{K}}$ which can be simplified as $\frac{\partial}{\partial\Gamma}\mathbf{D} = [\partial/\partial\gamma_k \mathbf{D}_{\gamma_k}]_{k\in\mathcal{K}}$. In this setting if the number of parameters in each γ_k is fixed, *e.g.* p, we can generate a parameter matrix $\Gamma_{p\times N}$ by putting γ_k's as the columns of $\Gamma_{p\times N}$. In the next section, it will be shown that such a setting can be used to generate a shift-resilient Gammatone parametric dictionary.

3. CASE STUDY: STRUCTURED GAMMATONE DICTIONARY

The Gammatone filterbanks have been shown to be closely related to the human auditory system [12] and the dictionary learned using audio training samples [13]. This model will be used here to find a reasonable size incoherent dictionary which has a shift-resilience structure for a more efficient dictionary implementation using filter banks. The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t), \tag{8}$$

where $B = f_c/Q + b_{min}$, f_c is the center frequency and a, b, Q, b_{min} and n are some constants. The dictionary is generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift. To induce the structure on the dictionary, let t_c be generated with a linear model, *i.e.* $t_c = t_0 + l\Delta : l \in \mathbb{N}_0$, where $t_0 \in [0, \Delta)$, Δ and $\mathbb{N}_0$ are the time-offset, the time-shift step size and non-negative integers. In this paper we assume that Δ is fixed during dictionary design, as letting Δ change, the PDD becomes very complicated. The difficulty is mainly caused by the fact that changing Δ can change the size of the dictionary. $\gamma_k = [t_{0k}\ f_{ck}\ n_k\ b_k]^T$ are thus the k^{th} optimization parameters. A set of atoms is generated using γ_k and $\{l : l \in \mathcal{L}\}$ followed by discretizing the atoms, see [4] for more detail on discretization. l is upperbounded such that t_c is always smaller than the atom length. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, Υ becomes a compact set. The parametric dictionary $\mathbf{D}_\Gamma$ is finally generated by concatenating $\mathbf{D}_{\gamma_k}$'s. The derivation of the dictionary with respect to Γ can be derived, using the structure explained in subsection 2.1, for each dictionary block $\mathbf{D}_{\gamma_k}$ similar to [4, Appendix B].

3.1. Simulation Results

The simulations are intended to first show the performance of the PDD algorithm in a structured setting, then demonstrate the advantages of designed dictionary in sparse approximation of audio signals using MP. The simulation parameters are presented in Table 1. The parameters in the first row and Δ are fixed and the others are used to generate the initial dictionaries, which might change throughout the PDD.

The simulations were run with two settings, 1) *unstructured:* no constraint on t_c and 2) *structured:* t_c follows the model $t_c = t_0 + l\Delta$. In the first experiment we designed the dictionary and showed the eigenvalues of the Gram matrices in Figure 1. The eigenvalues of a tight frame is also shown with a dashed line. Although the improvement of the eigenvalues, toward a tight frame, is not significant, it is changed in the right direction and is between the original and an unstructured parametric designed dictionary. The ℓ_2 norms of the columns of the Gram matrices of the mentioned dictionaries, which can show how much the corresponding atoms are correlated to the

Table 1. The parameters of the Structured PDD.

d	N	$\lvert\mathcal{K}\rvert$	b_{min}	Q	K	α
256	418	35	24.7	9.26	100	0.5

t_0	n	b	f_c	Δ
0	4	1	$50 + .27kB$	$\arg\max_t \lvert g(t)\rvert$

other atoms, are shown in the first row of Figure 2. The changes of the norms is obvious in the unstructured designed dictionary. To show that it is improved in the structured dictionary we also showed the reduction of the norms toward an ETF in the second row. Although the improvements in norms are small, most of the graph is in the positive orthant, which shows a reduction of the norm to a reference ETF.

Figures 1 and 2 show only a small achievement by structured PDD. This might be caused by selecting a highly restrictive structure for the dictionary. It is also relevant to investigate the performance of the structured parametric designed dictionary in sparse approximation of some sparsly structured signals. Some audio signals recorded from BBC Radio 3, which often plays classical music, have been used to evaluate the dictionaries. The average approximation errors, using 100 randomly selected audio samples, of the sparse approximations by applying MP algorithm are shown in Figure 3. The Structured dictionary shows a promising performance in this experiment.

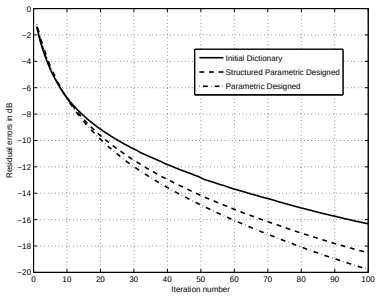

Fig. 3. Average approximation errors using MP for the initial, structured and unstructured designed dictionaries.

4. CONCLUSION

Imposing a structure on the parametric dictionary to facilitate the implementation of the designed dictionaries was investigated in this paper. A general form was introduced and a special case was investigated in more detail by using a case study. Another method was also presented to let the PDD be solved using conventional optimization techniques. Finally by some simulations on the Gammatone parametric dictionary, we showed that the designed dictionary is superior to the initial dictionary in sparse approximations of some selected audio signals.

One possible structure was explored in this report. There have been various structures introduced for dictionaries in dictionary learning problem. An independent research on these structures is left for future work. A structured parametric dictionary model can also be used in the dictionary learning problem. It preserves the structure of dictionary while adapting the dictionary to a given data.

The proposed algorithm for the parameter update needs to calculate the objective value in each iteration. It is a necessary step to guarantee the stability of the algorithm. Furthur investigations on the proposed algorithm might guarantee the stability of the overall algorithm without an explicit calculation of the objective.

5. REFERENCES

[1] J.E. Tropp and S.J. Wright, "Computational methods for sparse solution of linear inverse problems," Tech. Rep. 2009-01, California Institute of Technology, March 2009.

[2] M. Aharon, E. Elad, and A.M. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[3] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[4] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," accepted for publication in IEEE Transactions on Signal Processing, 2009.

[5] D.L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[6] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09)*, 2009.

[7] T. Strohmer and R.W. Heath, "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[8] J.A. Tropp, I.S. Dhillon, R.W. Heath Jr., and T. Strohmer, "Designing structural tight frames via an alternating projection method," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.

[9] S. Bonnabel and R. Sepulchre, "Geometric distance and mean for positive semi-definite matrices of fixed rank," Accepted for publication in SIAM Journal of Matrix Analysis. Available at `http://arxiv.org/abs/0807.4462`.

[10] B. Vandereycken, P.-A. Absil, and S. Vandewalle, "Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank," in *IEEE Workshop on Statistical Signal Processing*, Cardiff, UK, September 2009, pp. 389–392.

[11] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[12] R.D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the Gammatone function," Tech. Rep., APU Report, 1988.

[13] E.C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, pp. 978–982, 2006.

Dictionary Learning for Sparse Approximations with the Majorization Method

Mehrdad Yaghoobi, *Member, IEEE,* and Thomas Blumensath, *Member, IEEE* and Mike E. Davies, *Member, IEEE*

***Abstract*—In order to find sparse approximations of signals, an appropriate generative model for the signal class has to be known. If the model is unknown, it can be adapted using a set of training samples. This paper presents a novel method for dictionary learning and extends the learning problem by introducing different constraints on the dictionary. The convergence of the proposed method to a fixed point is guaranteed, unless the accumulation points form a continuum. This holds for different sparsity measures. The majorization method is an optimization method that substitutes the original objective function with a surrogate function that is updated in each optimization step. This method has been used successfully in sparse approximation and statistical estimation (e.g. Expectation Maximization (EM)) problems. This paper shows that the majorization method can be used for the dictionary learning problem too. The proposed method is compared with other methods on both synthetic and real data and different constraints on the dictionary are compared. Simulations show the advantages of the proposed method over other currently available dictionary learning methods not only in terms of average performance but also in terms of computation time.**

***Index Terms*—Dictionary Learning, Sparse Approximation, Majorization Methods, Surrogate Function Optimization Method, Block Relaxation Methods, Constrained Optimization**

I. Introduction

ORTHOGONAL function representations, introduced in the nineteenth century, are still a powerful tool in signal analysis. These representations have unique characteristics that make them suitable for many signal processing applications. In the last two decades, many researchers have tried to extend this idea to non-orthogonal and overcomplete representations [1], [2]. The overcomplete representation problem with the associated underdetermined linear system does not have a unique solution. The method of frames finds the minimum mean square solution and leads to representations where most of the coefficients are non-zero. Minimum mean square representations are desirable for some applications (e.g. robust transform coding in the presence of noise or erasure [3]) while there are other applications where sparsity of the representation is more desirable, e.g. in Compressed Sensing [4].

Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Institute for Digital Communication and with the Joint Research Institute for Signal and Image Processing, Edinburgh University, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, UK (e-mail: yaghoobi@ieee.org, thomas.blumensath@ed.ac.uk, mike.davies@ed.ac.uk).
This research was supported by EPSRC grant D000246/1. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the input signal and the coefficient vector respectively. The sparsest representation would be,

$$\min_x ||\mathbf{x}||_0 \quad s.t \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{1}$$

where $\mathbf{D}$ is a $d \times N$ matrix, often called *dictionary* and $||.||_0$ is the sparsity measure that counts the number of non-zero coefficients. This formulation can be relaxed to sparse *approximations* by using $||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \leq \epsilon$ with a small constant ϵ. Unfortunately finding the solutions to the above combinatorial problems is not easy in general [5]. Many approximations/relaxations have been presented to find acceptable solutions, e.g. [6], [7].

These methods are more successful at finding a sparse $\mathbf{x}$, when there is a suitable dictionary for the given signal. A simple method for dictionary generation is to add two or more orthogonal bases. Block-wise orthogonality can then be exploited to find the sparse approximation [8]. This also makes it easier to analyze the performance of sparse approximation methods [9], [10]. Another way to design a dictionary is to sample the parameters of an analytic function. For example a famous dictionary that has been used for overcomplete audio and image representations, is the Gabor dictionary [6]. These designed dictionaries are efficient when we have some a priori information about the signal's generative model. Alternatively it is possible to adapt the dictionary to a given source using a set of training samples ($\mathscr{Y} = \{\mathbf{y}^{(i)} : 1 \leq i \leq L\}$ where L is the number of training samples). Dictionary learning is the process of finding a dictionary $\mathbf{D}$ in which a given set of training samples has sparse representations (or approximations) $\mathscr{X} = \{\mathbf{x}^{(i)} : 1 \leq i \leq L\}$. Different methods have been proposed to learn dictionaries [11]–[15]. These methods are generally based on alternating minimization. In one step, a sparse approximation/representation algorithm finds sparse representations of the training samples with a fixed dictionary. In the other step, the dictionary $\mathbf{D}$ is updated to decrease the average approximation error while $\mathscr{X}$ (or the sparsity of $\mathscr{X}$ [15]) remains fixed. Because the objective functions are non-convex based on the pair of parameters ($\mathbf{D}$,$\mathscr{X}$), these methods generally only find a local minimum and different initial value for $\mathbf{D}$ (or $\mathscr{X}$), lead to different solutions. Nevertheless, in practice, good results have been reported [16], [17]. The proposed method in this paper uses a general formulation of alternating minimization. Therefore like other methods, we only expect to find local minima in general.

Contributions of the paper

This paper introduces a new algorithm for constrained dictionary learning which is very flexible and can use different constraints on the dictionary. The given method uses convex admissible sets whose boundaries are the same as the most frequently used admissible sets, however these convex sets allow the algorithm to generate a sequence throughout the sets (and not only on their boundaries). An advantage of the given method is that it optimizes a joint parameter objective function of the sparse coefficient matrix and the dictionary. In this framework, it is possible to choose a better path from the initial to the learnt dictionary by reducing the objective in different directions (coefficients or dictionary) in a cyclic way. This prevents oscillations of the sequence of updates around the optimal path and makes the algorithm more suitable for large scale problems, for which the calculation of sparse approximations of the training samples is often impossible.

Another advantage of the proposed algorithm is that we can impose a tighter constraint on the dictionary. For example, when a minimum size dictionary is required or when the optimum size of the dictionary is unknown, we can impose an additional penalty on the number of the atoms in the dictionary.

Numerical results show that the algorithm is faster than (or at least as fast as) most of the available dictionary learning methods.

Finally we show that the new algorithm is not only stable but also converges to a fixed point or its accumulation points form a continuum (in contrast to most of the dictionary learning methods, for which so far only stability has been shown).

Organization of the paper

An overview of previous dictionary learning methods is presented in Section II. Section III introduces the dictionary learning framework used in this paper. We introduce two new admissible sets for the dictionaries. Then, in Section III-A, we introduce the majorization method which is used in the matrix valued sparse approximation (III-B) and the dictionary update (III-C1, III-C2) steps. We introduce a new objective function to penalize the size of the dictionary in Section III-D. By minimization of the new objective function with the majorization minimization method, we find a minimum size dictionary. The different dictionary update methods are examined in the simulation section using training samples generated synthetically or sampled from an audio signal. After concluding the paper we present a convergence proof of the algorithm in Appendix B.

Notation

In this paper we use the following conventions. We use small and capital bold face characters for vector and matrix valued parameters respectively. In an iterative algorithm, the value of a parameter in the k^{th} iteration is distinguished by using the iteration number in square brackets, e.g. $\mathbf{D}^{[k]}$. We use a similar notation for a countable series. When a parameter appears with a hat, it shows the current value of that parameter. In the majorization method we introduce an auxiliary parameter which is distinguished with a double dagger superscript, e.g. $\mathbf{X}^{\ddagger}$. In dictionary learning, we have a set of training signals $\mathbf{y}^{(i)}$, where i is the signal index. Similarly, the associated coefficient vectors are $\mathbf{x}^{(i)}$. In this paper we use different norms for vectors and matrices. $||.||$ and $||.||_F$ are spectral and Frobenius norm in the Euclidean vector space respectively. $||.||_p : 0 < p \leq 1$ is the ℓ_p quasi-norm $(\sum |.|^p)^{\frac{1}{p}}$.

II. Dictionary Learning Methods

In traditional dictionary learning, one often starts with some initial dictionary and finds sparse approximations of the set of training signals while keeping the dictionary fixed. This is followed by a second step in which the sparse coefficients are kept fixed and the dictionary is optimized. This algorithm runs for a specific number of alternating optimizations or until a specific approximation error is reached. Most of these algorithms have been derived for dictionary learning in a noisy sparse *approximation* setting. Recently some researchers have considered dictionary learning for exact sparse representations [18], [19]. Like most other researchers, we consider dictionary learning for sparse approximation.

A. Sparse Approximation

Given a set of training samples $\mathbf{y}^{(i)}, \forall i : \ 1 \leq i \leq L$ and a dictionary $\mathbf{D}$, sparse approximations are often found by [1],

$$\begin{aligned} \mathbf{x}^{(i)*} &= \arg\min_{\mathbf{x}^{(i)}} \phi_i(\mathbf{x}^{(i)}) \ ; \\ \phi_i(\mathbf{x}) &= ||\mathbf{y}^{(i)} - \mathbf{D}\mathbf{x}||^2 + \lambda ||\mathbf{x}||_p^p \ , \ p \leq 1 \end{aligned} \tag{2}$$

An alternative to minimizing (2) individually on each vector is to find a joint sparse approximation of the matrix $\mathbf{Y} = [\mathbf{y}^{(1)} \ \mathbf{y}^{(2)} \ ... \ \mathbf{y}^{(L)}]$ by employing a sparsity measure in matrix form. The sparse matrix approximation problem can be formulated as,

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \phi(\mathbf{X}) \ ; \ \phi(\mathbf{X}) = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}), \tag{3}$$

where $J_{p,q}(\mathbf{X})$ is defined as [20],

$$J_{p,q}(\mathbf{X}) = \sum_{i \in I} [\sum_{j \in J} |x_{ij}|^q]^{p/q}. \tag{4}$$

For example, $||\mathbf{X}||_F = J_{2,2}^{1/2}(\mathbf{X})$ would be the Frobenius-norm. When $p = q$ all elements in $\mathbf{X}$ are treated equally.

In this paper we use $p = 1$, so that $J_{p,p}$ is convex. Extending the algorithm to $0 < p < 1$ is possible by using the majorization method proposed in [21]. However the convergence of the algorithm in this setting has not yet been proven [21], [22].

[1] Instead of minimizing an objective function like (2) one can also use a greedy algorithm. Because greedy algorithms do not deal with an objective function explicitly, convergence analysis of dictionary learning based on these methods is not easy and is therefore not considered here.

B. Dictionary Update

The second step in dictionary learning is the optimization of the dictionary based on the current sparse approximation. The cost function in (3) can be thought of as an objective function with two parameters,

$$\phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}). \quad (5)$$

Without additional constraints on the dictionary, minimizing the above objective function is an ill-posed problem. By constraining the norm of $\mathbf{D}$ we can solve the scale ambiguity[2] of the problem. Dictionaries with fixed column-norms or fixed Frobenius-norm have been used in different papers (for example [13] and [23]). We will use more general convex admissible sets defined in (7) and (8) below.

C. Previously Suggested Dictionary Update Methods

In the Method of Optimal Directions (MOD) [13] the best dictionary D is found using the pseudo inverse of X, followed by re-normalization of each atom. The Maximum Likelihood based Dictionary Learning algorithm [11], is similar to MOD but uses gradient optimization. In general, if the update is done iteratively, the best possible dictionary is typically calculated without any constraint. This update is then followed by normalization of the atoms. This normalization step can increase the total approximation error.

Kreutz-Delgado et al. [23] presented a dictionary learning method based on Maximum *a Posteriori* estimation (from now called MAP-DL[3]). By the use of an iterative method they estimate a dictionary that is consistent with a Bayesian model [23]. However, as reported in [15], when a fixed column-norm constraint is used, the algorithm updates atom by atom, making the method too slow for many applications.

The K-SVD method presented in [15] is fundamentally different from these methods. Instead of keeping the sparse coefficients fixed in the dictionary update step, only the support of the coefficient vectors (the positions of the non-zero coefficients) is kept fixed. Updates for each atom are found as the best normalized elementary function that matches the error (calculated after representing the signals with all atoms except the currently selected atom).

The formulation of the problem in this paper has several similarities with MAP-DL. However, our approach to solve this problem is based on a joint objective function for both the sparse approximation and the dictionary, which is good because we can develop a uniform approach for the updates and we have the flexibility to be able to switch between updating parameters easily. Furthermore, we use a different class of constraints on the desired dictionaries. In this setting, we will show a basic convergence proof. Our simulations furthermore show faster convergence for the proposed approach. Moreover, we can optimize the joint parameter objective function more wisely (see section III-E) and thereby increase the observed speed of convergence even further.

[2]Approximation error does not change by scaling up one parameter and scaling down the other one with the same scaling factor. Therefore the optimum $\mathbf{X}$ and $\mathbf{D}$ tend to zero and infinity respectively to minimize the sparsity penalty.

[3]Although MAP actually refers to an objective, MAP-DL is an algorithm for dictionary learning based on the MAP objective.

III. Dictionary Learning with the Majorization Method

We consider the dictionary learning problem as the following constrained optimization problem,

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X}} \phi(\mathbf{D},\mathbf{X}) \ s.t. \ \mathbf{D}\in\mathscr{D} \\ &\phi(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{p,p}(\mathbf{X}), \end{aligned} \quad (6)$$

where $\mathscr{D}$ is an admissible set of dictionaries. As noted in [23], two typical constraints are the unit Frobenius-norm and the unit column-norm constraints, both of which lead to non-convex solution sets. Instead of using these constraints in the algorithm derived below, we use the convex relaxed version of these constrained sets. These are the convex sets of matrices with bounded Frobenius norm,

$$\mathscr{D}_F = \{\mathbf{D}_{d\times N} : ||\mathbf{D}||_F \le c_F^{1/2}\} \quad (7)$$

where c_F is a constant and the convex set of matrices with bounded column norm,

$$\mathscr{D}_C = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 \le c_C^{1/2}\}, \quad (8)$$

where $\mathbf{d}_i$ is the i$^{\text{th}}$ column of the dictionary $\mathbf{D}$ and c_C is a constant. Note that when the sparsity measure in the sparse approximation step penalizes coefficients based on their magnitudes (e.g. $l_p : 0 < p \le 1$), it is easy to show that the solution of (6) is on the boundary of these convex admissible sets. However, the convex admissible sets also allow the optimization algorithm to "pass through" these admissible sets while the traditional non-convex sets only allow the algorithm to move along the boundary of these sets.

We use the block relaxation technique (see for example [24]) to solve (6), where $p = 1$, that is, in one step we fix $\mathbf{D}$ and minimize the objective based on $\mathbf{X}$, while in the other step we minimize the objective based on $\mathbf{D}$ with $\mathbf{X}$ fixed. This alternating minimization continues until the algorithm converges to an accumulation point. For a fixed dictionary, ℓ_1 penalized sparse approximation is a convex optimization problem and using convex dictionary admissible sets also turns the dictionary update into a convex optimization problem. Whilst this allows us to find the optimum update in each step, (5) is not convex as a function of the pair $(\mathbf{X},\mathbf{D})$, and alternating optimization is not guaranteed to find a global optimum.

Various methods have been presented to solve the ℓ_1 penalized sparse approximation [7], [25], [26]. We choose an Iterative Thresholding (IT) approach, which is a majorization minimization algorithm (see next subsection), which can be extended to the sparse approximation problem in matrix form (see III-B).

A. Majorization Minimization Method

Optimization of the problem in (6) with respect to any one of the parameters is challenging. We here use a technique called the "majorization method" [24], [27]. In the

majorization method, the objective function is replaced by a surrogate objective function which majorizes it and can be easily minimized. Here we are particularly interested in surrogate functions in which the parameters are decoupled, so that the surrogate function can be minimized element-wise.

A function ψ majorizes ϕ when it satisfies the following conditions,

$$\begin{aligned} &\phi(\omega) \leq \psi(\omega, \xi),\ \forall \omega, \xi \in \Upsilon \\ &\phi(\omega) = \psi(\omega, \omega),\ \forall \omega \in \Upsilon, \end{aligned} \tag{9}$$

where Υ is the parameter space. The surrogate function has an additional parameter ξ. At each iteration we first choose this parameter as the current value of ω and find the optimal update for ω.

$$\omega_{new} = \arg\min_{\omega \in \Upsilon} \psi(\omega, \xi) \tag{10}$$

We then update ξ with ω_{new}. The algorithm continues until we find an accumulation point. In practice the algorithm is terminated when the distance between ω and ω_{new} is less than some threshold.

This iterative method can be viewed as a block-relaxed minimization of the joint objective $\psi(\omega, \xi)$ [24]. In one step, we find the minimum of ψ based on ω. In the next step we minimize the objective based on ξ.

$$\xi_{new} = \arg\min_{\xi \in \Upsilon} \psi(\omega, \xi) \tag{11}$$

In our formulation, minimization of $\psi(\omega, \xi)$ based on ξ is done using $\xi_{new} = \omega$ (due to the definition of majorization in (9)). We use this interpretation of the majorization method to show the convergence of the proposed method in Appendix B.

There are different ways to derive a surrogate function. Jensen's inequality and Taylor series have often been used for this purpose [28] [29]. The Taylor series of a differentiable function $\phi(\omega)$ is,

$$\phi(\omega) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{1}{2!} d^2\phi(\xi)(\omega - \xi)^2 + o(\omega^3). \tag{12}$$

When ϕ has a bounded curvature ($d^2\phi < c_s$ for a finite constant c_s) this is majorized by,

$$\phi(\omega) \leq \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2, \forall \omega, \xi \in \Omega, \tag{13}$$

and we can define $\psi(\omega, \xi)$ (which satisfies (9)) as follows,

$$\psi(\omega, \xi) = \phi(\xi) + d\phi(\xi)(\omega - \xi) + \frac{c_s}{2}(\omega - \xi)^2. \tag{14}$$

Then, at each iteration, $\phi(\omega_{new}) \leq \psi(\omega_{new}, \omega) \leq \psi(\omega, \omega) = \phi(\omega)$, hence ϕ does not increase. Conditions for which these algorithms converge have been presented in [24] and [29]. The convergence of this method for sparse approximation is shown in [26]. A similar analysis can be derived for the iterative method in the dictionary update step.

In the next sections we show how we can use the majorization method to optimize the objective introduced in (6) based on $\mathbf{X}$ (Section III-B) or $\mathbf{D}$ (Sections III-C and III-D) using different constraints. Updating the coefficient or the dictionary matrices always reduces the joint objective function or keeps it at the same value. The fact that the objective function is lower-bounded is sufficient to show stability of the updating process in the sense of Lyapunov (Lyapunov second theorem) [30]. We also provide a basic convergence proof for the proposed algorithm in Appendix B.

Algorithm 1 : $\mathcal{SA}(\mathbf{X}_t, \mathbf{D}_t)$

initialization: $c_X > ||\mathbf{D}_t^T \mathbf{D}_t||$, $\mathbf{X}^{[0]} = \mathbf{X}_t$
for $n = 1$ **to** K_X **do**
  $\mathbf{A} = \frac{1}{c_X}(\mathbf{D}_t^T \mathbf{Y} + (c_X \mathbf{I} - \mathbf{D}_t^T \mathbf{D}_t)\mathbf{X}^{[n-1]})$
  $\mathbf{X}^{[n]} = \mathcal{S}_\lambda(\mathbf{A})$
end for
output: $\mathbf{X}_{t+1} = \mathbf{X}^{[K_X]}$

B. Matrix Valued Sparse Approximation

We begin by showing how the majorization method is used for the first step of the alternating minimization: matrix valued sparse approximation. The updating formula derived here is used in the generalized block relaxation method derived later in this section. For fixed $\mathbf{D}$, we use the matrix form of the Taylor series inequality (13), see Appendix A, to derive the following majorizing function,

$$\begin{aligned} ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 &\leq ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 \\ &\quad + c_X ||\mathbf{X} - \mathbf{X}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X} - \mathbf{D}\mathbf{X}^{[n-1]}||_F^2 \\ &= ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 + \pi_{\mathbf{X}}(\mathbf{X}, \mathbf{X}^{[n-1]}) \end{aligned} \tag{15}$$

where $\mathbf{X}^{[n-1]}$ is the coefficient matrix in the previous step, $\pi_{\mathbf{X}}(\mathbf{X}, \mathbf{X}^{[n-1]}) := c_X ||\mathbf{X} - \mathbf{X}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X} - \mathbf{D}\mathbf{X}^{[n-1]}||_F^2$ and $c_X > ||\mathbf{D}^T\mathbf{D}||$ is a constant, where $||.||$ is defined as the spectral norm [31]. This type of majorization has already been used for sparse approximation with vector valued coefficients [26], [32], [33]. $\Phi(\mathbf{D}, \mathbf{X})$ in (6) has two terms, $||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2$ and $\lambda J_{p,p}(\mathbf{X})$. Therefore a function majorizing $\Phi(\mathbf{D}, \mathbf{X})$ is,

$$\Phi(\mathbf{D}, \mathbf{X}) \leq \Phi(\mathbf{D}, \mathbf{X}) + \pi_{\mathbf{X}}(\mathbf{X}, \mathbf{X}^{[n-1]}) \tag{16}$$

Let $\mathbf{A} := \frac{1}{c_X}(\mathbf{D}^T\mathbf{Y} + (c_X \mathbf{I} - \mathbf{D}^T\mathbf{D})\mathbf{X}^{[n-1]})$. It can be shown that the optimum of the surrogate objective (16), where $p = 1$, is found by shrinking elements in $\mathbf{A}$ [26], [34], that is,

$$\{\mathbf{X}^{[n]}\}_{i,j} = \mathcal{S}_\lambda(\mathbf{A}) = \begin{cases} a_{i,j} - \lambda/2\ sign(a_{i,j}) & \lambda/2 < |a_{i,j}| \\ 0 & otherwise. \end{cases} \tag{17}$$

The matrix $\mathbf{A}$ is the modified *Landweber update* [35], (which is a gradient descent update) of the matrix valued coefficients. This iterative update continues until $\mathbf{X}^{[n]}$ converges to the optimum solution. The pseudocode for this coefficient update is presented in Algorithm 1. The operator $\mathcal{S}_\lambda$ is the shrinkage operator defined in (17).

C. Dictionary Update

In the second step of the alternating minimization, we minimize the objective function with respect to $\mathbf{D}$ keeping $\mathbf{X}$ fixed. This constrained minimization problem can be solved using several methods. Among these, fixed-point iteration and

iterative gradient projection methods have been suggested for the dictionary updates in [23], [11]. In this paper we derive a majorization method for the dictionary update.

The quadratic part of the objective function in (6) has a bounded curvature when minimizing over $\mathbf{D}$. So again using the Taylor series, the majorizing function is as follows,

$$\begin{aligned}||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 \leq\ & ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 \\ & + c_D||\mathbf{D}-\mathbf{D}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^{[n-1]}\mathbf{X}||_F^2 \\ & = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]})\end{aligned} \tag{18}$$

where $\mathbf{D}^{[n-1]}$ is the dictionary found in the previous step, $\pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}) := c_D||\mathbf{D}-\mathbf{D}^{[n-1]}||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^{[n-1]}\mathbf{X}||_F^2$ and $c_D > ||\mathbf{X}^T\mathbf{X}||$ is a constant. When $\mathbf{X}$ changes in the sparse approximation step, this spectral norm needs to be re-calculated. We know that the spectral norm of a Hermitian matrix is its largest eigenvalue and various efficient methods have been presented to calculate it [36].

This majorizing function can be used with different constraints. In the following two subsections we derive the optimum of (18) under bounded Frobenius and column-norm constraints.

1) Constrained Frobenius-Norm Dictionaries: An advantage of using a constraint on the Frobenius-norm of the dictionary is that the learnt dictionary can have columns with different norms. Such dictionaries can then be used in the weighted-pursuit framework [37], where atoms with large norms have more chance to appear in the approximations. It has been shown that the average performance of sparse approximation increases when the weights are chosen correctly for the class of signals under study [37].

In the dictionary update step, with the help of a Lagrangian multiplier γ, we turn (6) into an unconstrained optimization problem,

$$\min_{\mathbf{D}} \phi_\gamma(\mathbf{D},\mathbf{X}), \tag{19}$$

where $\phi_\gamma(\mathbf{D},\mathbf{X})$, for $p=1$, is now defined as,

$$\phi_\gamma(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + \gamma(||\mathbf{D}||_F^2 - c_F). \tag{20}$$

Fixing $\mathbf{X}$, the solution to this minimization problem is a global minimum if the solution satisfies the K.K.T conditions [38, Theorem 28.1]. As the admissible set is convex, any minimum of $\Phi_\gamma(\mathbf{D},\mathbf{X})$ is an optimal solution if $\gamma(||\mathbf{D}||_F^2 - c_F) = 0$. Therefore if $||\mathbf{D}||_F^2 \neq c_F$, γ must be zero.

The majorizing function is generated by adding $\pi_{\mathbf{D}}$ to the objective function,

$$\psi_\gamma(\mathbf{D},\mathbf{D}^{[n-1]}) = \phi_\gamma(\mathbf{D},\mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}). \tag{21}$$

$\mathbf{X}$ has here been omitted from the list of parameters because it is assumed fixed in the dictionary update step. The optimum of this function is at a point with zero gradient,

$$\begin{aligned}\frac{d}{d\mathbf{D}}\psi_\gamma(\mathbf{D},\mathbf{D}^{[n-1]}) = & -2\mathbf{X}\mathbf{Y}^T + 2\mathbf{X}\mathbf{X}^T\mathbf{D}^{[n-1]^T} + 2c_D\mathbf{D}^T \\ & - 2c_D\mathbf{D}^{[n-1]^T} + 2\gamma\mathbf{D}^T = \mathbf{0}\end{aligned}$$

Algorithm 2 : $\mathcal{DU}(\mathbf{X}_{t+1},\mathbf{D}_t)$

1: **initialization:** $c_D > ||\mathbf{X}_{t+1}^T\mathbf{X}_{t+1}||$, $\mathbf{D}^{[0]} = \mathbf{D}_t$
2: **for** $n = 1$ **to** K_D **do**
3: $\quad \mathbf{B} = \frac{1}{c_D}(\mathbf{Y}\mathbf{X}_{t+1}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}_{t+1}\mathbf{X}_{t+1}^T))$
4: $\quad \mathbf{D}^{[n]} = \mathcal{P}(\mathbf{B})$
5: **end for**
6: **output:** $\mathbf{D}_{t+1} = \mathbf{D}^{[K_D]}$

By solving the above equation we find the optimal dictionary,

$$\mathbf{D}_\gamma^* = \frac{c_D}{\gamma + c_D}\mathbf{B} \tag{22}$$

where $\mathbf{B}$ is defined as

$$\mathbf{B} := \frac{1}{c_D}(\mathbf{Y}\mathbf{X}^T + \mathbf{D}^{[n-1]}(c_D\mathbf{I} - \mathbf{X}\mathbf{X}^T)). \tag{23}$$

$\mathbf{B}$ has again the same role as the *Landweber update*. To satisfy the K.K.T. conditions, a non-negative γ has to be found such that $\gamma(||\mathbf{D}^{[n]}||_F^2 - c_F) = 0$. If $\mathbf{D}_0^* = \mathbf{B}$ is admissible, we can update the dictionary $\mathbf{D}^{[n]} = \mathbf{B}$. Otherwise we scale $\mathbf{B}$ to have Frobenius-norm equal to $c_F^{1/2}$.

$$\mathbf{D}^{[n]} = \mathcal{P}_{c_F}^F(\mathbf{B}) = \begin{cases}\mathbf{B} & ||\mathbf{B}||_F \leq c_F^{1/2} \\ \frac{c_F^{1/2}}{||\mathbf{B}||_F}\mathbf{B} & otherwise\end{cases} \tag{24}$$

The pseudocode for this dictionary update is presented in Algorithm 2. Here $\mathcal{P}$ is the operator $\mathcal{P}_{c_F}^F$ presented in (24). In the following, we show that the dictionary updates, subject to the constraints on the column-norms or the joint sparsity (see below) of the dictionaries, have similar algorithms, but with the different operators for $\mathcal{P}$.

If we use an equality in the definition of (7), i.e. we demand a *fixed* Frobenius-norm, γ can become negative. In this case the decision criteria of (24) becomes an equality ($||\mathbf{B}||_F = c_F^{1/2}$).

2) Constrained Column-Norm Dictionaries: Another often used admissible set in dictionary learning is the set of *fixed* or unit column norm matrices. Instead a bound on the column norms of the dictionary can be used to get a convex admissible set. To make (6) an un-constrained optimization problem we need N Lagrangian multipliers (equal to the number of constraints),

$$\min_{\mathbf{D}} \phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}), \tag{25}$$

where $\phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X})$, for $p=1$, is now defined as,

$$\phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + \sum_{i=1}^{N}\gamma_i(\mathbf{d}_i^T\mathbf{d}_i - c_C) \tag{26}$$

With this formulation, the K.K.T conditions are,

$$\forall i : 1 \leq i \leq N, \qquad \gamma_i(\mathbf{d}_i^T\mathbf{d}_i - c_C) = 0\ . \tag{27}$$

This means that for each i when $\mathbf{d}_i^T\mathbf{d}_i$ is not equal to c_C, γ_i should be zero. (25) can be rewritten as

$$\phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2 + \lambda J_{1,1}(\mathbf{X}) + tr\{\mathbf{\Gamma}(\mathbf{D}^T\mathbf{D} - c_C\mathbf{I})\}, \tag{28}$$

where Γ is a diagonal matrix with the γ_i as the i^{th} diagonal element. By adding $\pi_{\mathbf{D}}$, we get the majorizing function,

$$\psi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{D}^{[n-1]}) = \phi_{\mathbf{\Gamma}}(\mathbf{D},\mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}). \quad (29)$$

The gradient is again set to zero and the optimum solution is found to be,

$$\mathbf{D}^*_{\Gamma} = \mathbf{B}(\frac{1}{c_D}\mathbf{\Gamma}+\mathbf{I})^{-1}, \quad (30)$$

where $\mathbf{B}$ has the same definition as introduced in (23). All γ_i are non-negative and $(\frac{1}{c_D}\mathbf{\Gamma}+\mathbf{I})$ is an (invertible) diagonal matrix. In equation (30), by changing γ_i, we multiply the corresponding column of $\mathbf{B}$ by a scalar. We start by setting all $\gamma_i = 0$. For any columns of $\mathbf{D}^*_{\mathbf{0}} = \mathbf{B}$ for which the norm is more than $c_C^{1/2}$, we find the smallest value of γ_i which scales down that column to have the largest acceptable norm ($c_C^{1/2}$).

$$\begin{aligned}\mathbf{D}^{[n]} &= \mathcal{P}^C_{c_C}(\mathbf{B}) = \{\mathbf{b}_j^{[n]}\}_{1\le j\le N}\\ \mathbf{d}_j^{[n]} &= \begin{cases}\mathbf{b}_j & ||\mathbf{b}_j||_2 \le c_C^{1/2}\\ \frac{c_C^{1/2}}{||\mathbf{b}_j||_2}\mathbf{b}_j & otherwise,\end{cases}\end{aligned} \quad (31)$$

where $\mathbf{d}_j$ and $\mathbf{b}_j$ are the j^{th} columns of $\mathbf{D}$ and $\mathbf{B}$ respectively.

Alternatively, we can use a *fixed* column-norm constraint ($\mathscr{D} = \{\mathbf{D}_{d\times N} : ||\mathbf{d}_i||_2 = c_C^{1/2}\}$). Here the algorithm may find a Γ in which some of the γ_i are negative. The dictionary update can then be found by a similar operator as (31) but with equality in the decision criteria ($||\mathbf{b}_j||_2 = c_C^{1/2}$) or simply by

$$\mathbf{d}_j^{[n]} = \frac{c_C^{1/2}}{||\mathbf{b}_j||_2}\mathbf{b}_j. \quad (32)$$

When the norm of any columns of $\mathbf{B}$ is zero, we have some ambiguity in the update formula. In this case we can shrink the size of the dictionary by deleting this atom or keep the size fixed by introducing a random atom to the dictionary. In practice we have not encountered such an ambiguity.

D. Jointly Sparse Dictionaries

The majorization approach to dictionary learning is extremely flexible. To demonstrate this, we introduce an additional constraint that encourages dictionary size reduction. In some applications there is a benefit in using a smaller dictionary. One of these benefits could be in coding, where the coding cost increases when the size of the dictionary grows. To shrink the dictionary size during learning, we introduce the following additional constraint on the number of atoms in the dictionary.

$$\min_{\mathbf{X},\mathbf{D}\in\mathscr{D}} \phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) \ ;$$

$$\phi_{\theta,0,\infty}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2+\lambda J_{1,1}(\mathbf{X})+\theta||\max_i|\{\mathbf{D}\}_{i,j}||_0$$

where $||.||_0$ is an operator that counts the number of non-zero elements. Because $\phi_{\theta,0,\infty}$ is non-convex and non-continuous, we replace the objective function with a relaxed version as follows,

$$\min_{\mathbf{X},\mathbf{D}\in\mathscr{D}} \phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) \ ;$$

$$\phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) = ||\mathbf{Y}-\mathbf{D}\mathbf{X}||_F^2+\lambda J_{1,1}(\mathbf{X})+\theta J_{1,q}(\mathbf{D}^T) \quad (33)$$

This objective is convex when $\mathbf{X}$ is fixed. For fixed $\mathbf{X}$, to minimize over $\mathbf{D}$, the joint sparsity penalty is again decoupled by adding $\pi_{\mathbf{D}}$, (defined above), to the objective function

$$\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) = \phi_{\theta,1,q}(\mathbf{D},\mathbf{X}) + \pi_{\mathbf{D}}(\mathbf{D},\mathbf{D}^{[n-1]}). \quad (34)$$

By separating the terms depending on $\mathbf{D}$, the surrogate cost can be written as,

$$\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) \propto c_s tr\{\mathbf{D}\mathbf{D}^T - 2\mathbf{B}\mathbf{D}^T\} + J_{1,q}(\mathbf{D}^T) \quad (35)$$

where $\mathbf{B}$ is defined in (23). The dictionary constraint is again introduced into the objective function using Lagrangian multiplier(s). Let $\mathbf{d}_j$ and $\mathbf{b}_j$ be the j^{th} columns of $\mathbf{D}$ and $\mathbf{B}$ respectively. The objective function, using the bounded column-norm (8), can be written as,

$$\begin{aligned}\psi_{\theta,1,q}(\mathbf{D},\mathbf{D}^{[n-1]}) &\propto \sum_j (tr\{\tau_j^2\mathbf{d}_j\mathbf{d}_j^T - 2\mathbf{b}_j\mathbf{d}_j^T\} + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q)\\ &= \sum_j (\tau_j^2\mathbf{d}_j^T\mathbf{d}_j - 2\mathbf{d}_j^T\mathbf{b}_j + \tfrac{\theta}{c_D}||\mathbf{d}_j||_q)\\ &\propto \sum_j ((\tau_j\mathbf{d}_j - \mathbf{b}_j/\tau_j)^2 + \tfrac{\theta}{c_D\tau_j}||\tau_j\mathbf{d}_j||_q)\\ &= \sum_j \psi_q^{\frac{\theta}{c_D\tau_j}}(\tau_j\mathbf{d}_j, \mathbf{b}_j/\tau_j)\end{aligned} \quad (36)$$

where $\psi_q^\alpha(\mathbf{v},\mathbf{w}) = (\mathbf{w}-\mathbf{v})^2 + \alpha||\mathbf{v}||_q$, $\tau_j = (1+\gamma_j/c_D)^{1/2}$ and γ_j are the Lagrangian multipliers. To minimize (36), we can minimize the first term by minimizing ψ_q^α for each $\mathbf{d}_j$ independently. With the help of two lemmas presented in [39], we can find the optimum of ψ_q^α based on $\mathbf{d_j}$ for $q = 1, 2$ and ∞. The minimum of $\psi_q^\alpha(\mathbf{v},\mathbf{w})$ based on $\mathbf{v}$ [39, Lemma 4.1] is,

$$\min_{\mathbf{v}} \psi_q^\alpha(\mathbf{v},\mathbf{w}) = \mathbf{w} - \mathscr{P}_\alpha^{q'}(\mathbf{w}) \quad (37)$$

where $\mathscr{P}_\alpha^{q'}$ is the orthogonal projection onto the dual norm ball with radius $\mathbf{w}$ and the dual norm is defined as $||.||_{q'}$ with $1/q' + 1/q = 1$. This minimization problem can be solve analytically for some q [39, Lemma 4.2]. In this paper we derive the dictionary update formula for $q = 2$. Interested readers can derive the update formulas when $q = 1$ or $q = \infty$ in the same way. We have

$$\begin{aligned}\mathbf{B}^*_\tau &= \{\mathbf{b}_j^*\}_{1\le j\le N}\\ \mathbf{b}_j^* &= \arg\min_{\mathbf{d}_j} \psi_2^{\frac{\theta}{c_s\tau_j}}(\tau_j\mathbf{d}_j, \mathbf{b}_j/\tau_j)\\ &= \begin{cases}\frac{1}{\tau_j^2}(1-\frac{\theta}{2c_D||\mathbf{b}_j||_2})\ \mathbf{b}_j & \frac{\theta}{2c_D} < ||\mathbf{b}_j||_2\\ 0 & otherwise\ ,\end{cases}\end{aligned} \quad (38)$$

where $\tau = \{\tau_j\}_{1\le j\le N}$. When all γ_j are non-negative, for any inadmissible $\mathbf{b}_j^*$ with $\tau_j = 1$ ($\gamma_j = 0$), one can decrease $||\mathbf{d}_j^*||_2$ to $c_C^{1/2}$ by increasing τ_j to satisfy the K.K.T conditions. Let $\mathcal{S}^J_{\frac{\theta}{c_D}}(\mathbf{B}) := \mathbf{B}^*_{\tau=\mathbf{1}}$ for any $\mathbf{B}$ found by (23). The dictionary update is therefore done by $\mathcal{P}^C_{c_C}\mathcal{S}^J_{\frac{\theta}{c_D}}(\mathbf{B})$.

When we are looking for a bounded Frobenius-norm dictionary, the dictionary update could be derived, using a similar approach, by $\mathcal{P}^F_{c_F}\mathcal{S}^J_{\frac{\theta}{c_D}}(\mathbf{B})$.

Algorithm 3 : $\mathcal{DL}(\mathbf{X}_0, \mathbf{D}_0)$

1: **for** $t = 1$ **to** T **do**
2: $\quad \mathbf{X}_{t+1} = \mathcal{SA}(\mathbf{X}_t, \mathbf{D}_t)$
3: $\quad \mathbf{D}_{t+1} = \mathcal{DU}(\mathbf{X}_{t+1}, \mathbf{D}_t)$
4: **end for**
5: **output:** $\mathbf{D}_T$

E. Generalized block relaxation method for dictionary learning

In the previous subsections we presented a block relaxation method to optimize $\mathbf{X}$ and $\mathbf{D}$ iteratively. In each step, we used an iterative method to find the optimum solution based on one variable while keeping the other variable fixed. The pseudocode for dictionary learning in this framework is presented in Algorithm 3.

Because the joint objective function does not have a fixed bounded curvature, we could not use the majorization method for both parameters jointly. On the other hand, this alternating optimization decreases the rate of convergence as it often oscillates around the optimal path. Instead of fully optimizing with respect to a single parameter in each step, the generalized block relaxation method updates each variable at a time and reduces the objective function, using for example a cyclic selection or any other periodic selection of the parameters. A simple way to choose which parameter to update is to calculate the update based on each parameter and then choose the parameter that decrease the objective function the most. A drawback of this type of parameter selection is that it doubles the computational cost. Another technique is to alternatively update each parameter. For dictionary learning, we found that using more coefficient updates than dictionary updates is in general more beneficial. So one can use p updates of $\mathbf{X}$ followed by q updates of $\mathbf{D}$ when $p \geq q$.

A more complete explanation and a basic convergence proof for the generalized block relaxed dictionary learning algorithm are provided in Appendix B. It is easy to show that the block relaxation method is a special case of the generalized block relaxation method. Therefore convergence of the block relaxation method (alternating minimization) for the dictionary learning follows as a corollary of this result.

IV. SIMULATIONS

We evaluate the proposed method with synthetic and real data. Using synthetic data with random dictionaries helps us to examine the ability of the proposed methods to recover dictionaries exactly (to within an acceptable squared error). We generated the synthetic data and dictionaries as proposed in [23] and [15]. To evaluate the performance on real data, we chose audio signals, which have been shown to have some sparse structure. We then used the learnt dictionary for audio coding and show some improvements in Rate-Distortion performance compared to coding with classical dictionaries.

A. Synthetic Data

A 20×40 matrix $\mathbf{D}$ was generated by normalizing a matrix with i.i.d. uniform random entries. The number of non-zero elements in each of the coefficient vectors was selected between 3 and 7. The locations of the non-zero coefficients were selected uniformly at random. We generated 1280 training samples where the absolute values of the non-zero coefficients were selected uniformly between 0.2 and 1. In the setting for exact dictionary recovery [15], [23] and under a mild condition, the constrained column-norm dictionary and the K-sparse signals are the global solutions of the dictionary learning problem based on exact sparse representations and the ℓ_1 based exact sparse representation problems, respectively (see for example [19]). The proposed algorithm as well as the other dictionary learning algorithms discussed, are proposed for sparse *approximations*, that is, they allow approximation error when calculating the sparse coefficients. To adapt the algorithm to this problem, we assumed that the sparse approximation finds the correct support in each step. Once the support has been identified, we find the best approximation by projecting onto the selected sub-space. This is called debiasing.

We here compare the majorization based dictionary learning

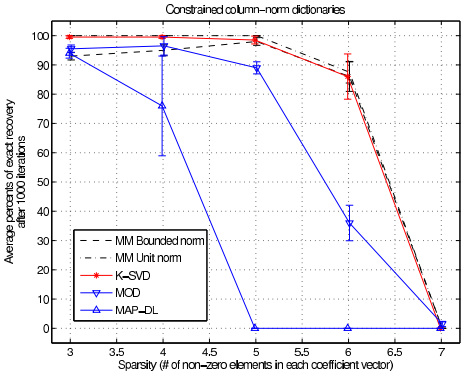

Fig. 1. A comparison of the dictionary recovery success rates using different dictionary learning methods under a column-norm constraint.

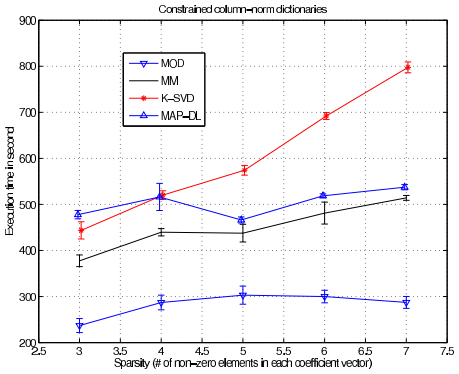

Fig. 2. A comparison of the computation costs of the dictionary learning methods under a column-norm constraint.

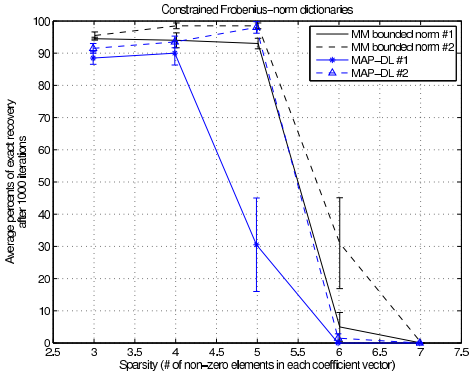

Fig. 3. A comparison of the dictionary recovery success rates using MM and MAP dictionary learning methods under a Frobenius norm constraint: 1: Desired dictionary had fixed Frobenius-norm. 2: Desired dictionary had fixed column-norms.

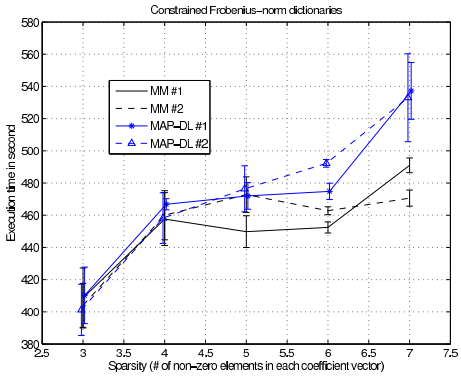

Fig. 4. A comparison of the computation costs of the dictionary learning methods under a Frobenius norm constraint.

algorithm to MOD, K-SVD and MAP-DL. The stopping criteria for IT was the distance between two consecutive iterations ($\delta = 3\times10^{-4}$) and λ was set to 0.4. The termination conditions for the iterative dictionary learning methods (majorization method for dictionary learning (MM-DL) and MAP-DL) was set to ($||\mathbf{D}^{[n]} - \mathbf{D}^{[n-1]}||_F \leq 10^{-7}$).

We started from a normalized random $\mathbf{D}$ and used 1000 iterations. The learning parameter (γ) in MAP-DL was selected as described in [23] and we down-scaled γ by a factor of 2^{-j} ($j > 1$) when the algorithm was diverging. To allow a fair comparison, we repeated the simulations 5 times. If the squared error between a learnt and true dictionary element was below 0.01, it was classified as correctly identified. The average percentages and standard deviations are shown in Figure 1. It can be seen that in all cases, MM-DL with fixed column-norm and K-SVD recovered nearly the same number of atoms and performed better than the other methods (although, for the signals with less than 6 non-zero coefficients, MM-DL recovered all desired atoms, performance of K-SVD was very close to it). The debiasing process creates some ambiguities in dictionary learning when using the bounded-norm constraints as they reduce the effect of the coefficient magnitudes in the sparsity measure. Therefore, we observe atoms which do not have a boundary norm (here, unit norm), even after 1000 iterations. In this case, we get better results using a fixed column-norm admissible set which resolves this ambiguity. The MAP-DL algorithm did not perform well in this simulation. We guess the reason for this is slow convergence of the approach and the use of more iterations might improve the performance.

In Fig.2 we compare the computation time of the algorithms for the above simulations. Simulations ran on the Intel Xeon 2.66 GHz dual-core processor machine and both cores were used by Matlab. In this graph the total execution time of the algorithms (sparse approximations plus dictionary updates for 1000 iterations) is shown. MOD was fastest followed by our MM-DL.

We have a larger admissible set when fixing the Frobenius-norm of the dictionary, which makes the problem of exact recovery more complicated and we expect to observe worse performance in terms of exact atom recovery. To test this, we started with a normalized random dictionary, normalized either to have fixed Frobenius-norm or fixed column-norm. The simulations were repeated for 5 trials and the averages and standard deviations of the atom recovery are shown in Fig. 3. In these simulations MM-DL performed slightly better than MAP-DL. The other observation in this figure is that when the desired dictionaries have equal column-norms, performance of the algorithms increase but do not reach the performance observed when using the more restricted (and appropriate) admissible set. Computation times of the algorithms, on the machine described formerly, are shown in Fig.4.

In the next experiment we assume that the desired dictionary size is unknown but bounded. We generated the data as in the previous experiments but the simulations were started with four times overcomplete dictionaries (two times larger than the desired dictionary size). The dictionary updates were based on the joint sparsity objective function (33) (with $\theta = 0.05$, $p = 1$ and $q = 2$). The average percentage of exact atom recovery for 5 trials are shown in Fig. 5 and 6. We plotted the percentage of the exact recovery of the original atoms, regardless of the learnt dictionary size. In the lower plot, we show the size of dictionary after 1000 iterations. With this θ we identified the size correctly but for less sparse signals (higher k) we got less accurate results. The overall performance of the algorithm is determined by the correct choice of θ. By increasing θ we find smaller dictionaries and vice versa.

B. Dictionary Learning for Sparse Audio Coding

In this section we demonstrate the performance of the proposed dictionary learning method on audio signals and thus show that our method is applicable to large dictionary learning problems. An audio sample of more than 8 hours was recorded from BBC radio 3, which plays mostly classical music.

In the first experiment we used bounded column-norm and bounded Frobenius-norm dictionary admissible sets. The audio

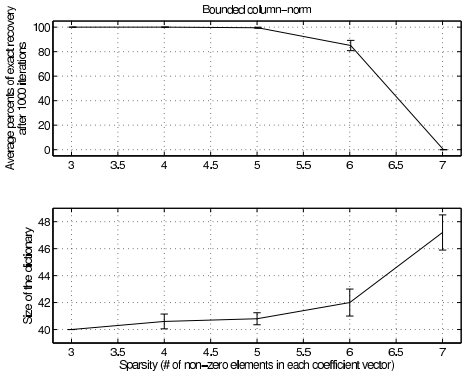

Fig. 5. Dictionary recovery success rates under a column-norm constraint and joint sparsity penalty.

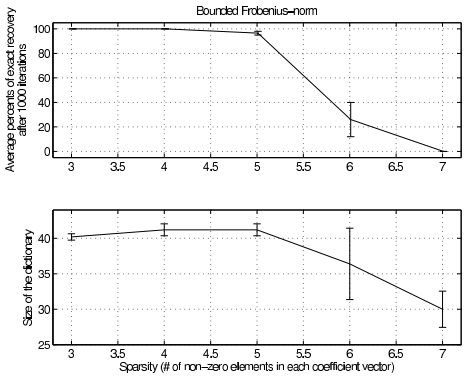

Fig. 6. Dictionary recovery success rates under a Frobenius norm constraint and joint sparsity penalty.

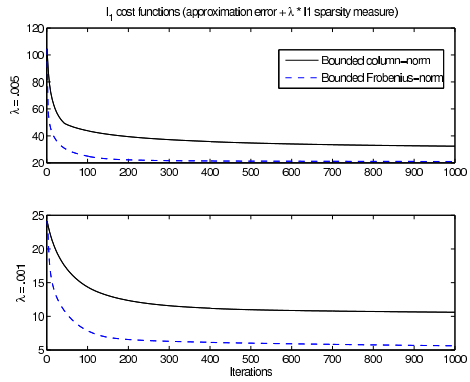

Fig. 7. ℓ_1 cost functions for two different Lagrangian multipliers (λ) .005 (top) and .001 (bottom).

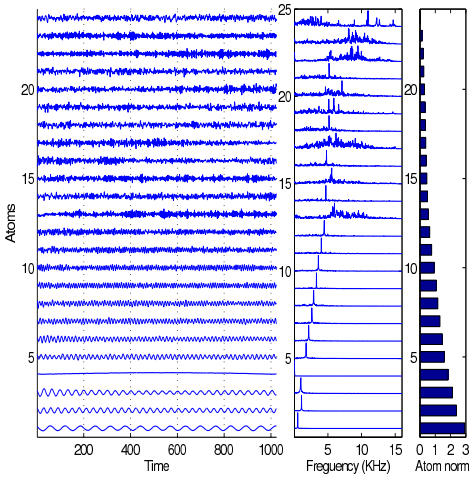

Fig. 8. A selection of learnt atoms in time (left) and frequency (middle) domain. Their norms are shown in the right panel.

sample was summed to mono and down-sampled by a factor of 4. From this 12kHz audio signal, we randomly took 4096 blocks of 256 samples each. The set of dictionaries with the column-norms bounded by c_C is a subset of the set of bounded Frobenius-norm dictionaries, when $c_F = Nc_C$. We chose dictionary admissible sets with column-norms and Frobenius-norms bounded by $c_C = 1$ and $c_F = N$ respectively. We initialized the dictionary with a 2 times overcomplete random dictionary and used 1000 iterations. The objective function against iteration, for two different values of λ, are shown in Fig. 7. This figure shows that the optimal bounded Frobenius-norm dictionaries are better solutions for the objective functions.

As a second experiment, we looked at an audio coding example. We used the proposed method with the bounded Frobenius-norm constraint to learn a dictionary based on a training set of 8192 blocks, each 1024 samples long. In this experiment we want to learn the dictionary for a larger block length than the previous experiment. The convergence of the traditional block relaxation method for a problem with this size is very slow. Therefore we run the simulations with the generalized block relaxation method and a joint sparsity constraint on the dictionary to encourage shrinkage of the dictionary. This shrinkage makes the algorithm faster in later iterations. Even though the recorded audio had 48k samples per second, the audio had a maximum frequency of 16kHz. Therefore we downsampled the original audio by a factor of 3/2 without any degradation in the audio fidelity. It has been shown that audio can be modeled reasonably well using tonal, transient and noisy residual components [40]. We chose a 2 times overcomplete sinusoid dictionary (frequency oversampled DCT) as the initialization point and ran the simulations with different lambda values for 5000 iterations of alternative optimization of (41), which took approximately 8 hours for each λ, running on the machine mentioned in the previous

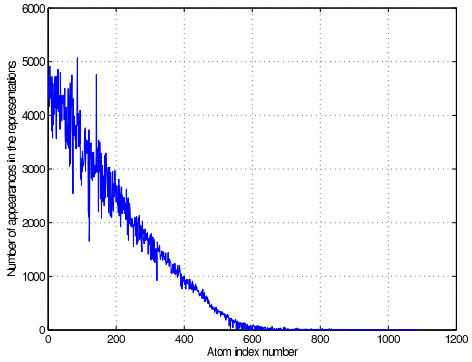

Fig. 9. Number of appearances of the learnt atoms in the representations of the training samples (of size 8192).

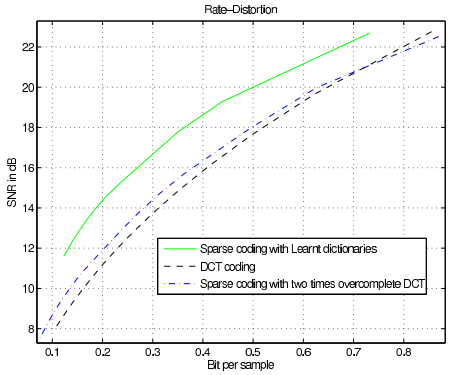

Fig. 10. Estimated Rate-Distortion for the audio coding example using the learnt dictionary, the shrunk 2 times overcomplete DCT dictionary and the DCT.

subsection.

A subset of the learnt atoms ($\lambda = .01$, $\theta = .01$), which is selected by uniformly sampling the atom indices, is shown in Fig. 8. These atoms are shown in the time and frequency domain in the left and middle windows respectively. The norms of the selected atoms are shown in the right window. The number of appearances of each atom, which are sorted based on their ℓ_2 norms, are shown in Fig. 9. To design an efficient encoder we only used atoms that were used frequently in the representations. Therefore we were able to further shrink the dictionary size. In this test we chose a threshold of 40 appearances (out of 8192) as the selection criteria. This dictionary was used to find the sparse approximations of 4096 different random blocks, each of 1024 samples, from the same data set. We then encoded the location (significant bit map) and magnitude of the non-zero coefficients separately. In this paper we used a uniform scalar quantizer with a double zero bin size to code the magnitude. We estimated the entropy of the coefficients to approximate the required coding cost. To encode the significant bit map, we assumed an i.i.d. distribution for the location of the non-zero atoms. The same coding strategy was used to code sparse approximations with a two times frequency overcomplete DCT (the initial dictionary used for learning) followed by shrinking based on the number of appearances. For reference we calculated the rate-distortation of the DCT coefficient encoding of the same data, using the same method of significant bitmap and non-zero coefficients coding. The performance is compared in Fig. 10. In the sparse coding methods, the convex hulls of the rate-distortion performances calculated with different dictionaries, each optimized and shrunk for different bit-rate, are shown in this figure. Using the learnt dictionaries for sparse approximation is superior to using the DCT or overcomplete DCT for the range of bit-rates shown.

It would be nice to compare these real data experiments with K-SVD, which is shown to perform well in dictionary learning for medium size problems. However, we found K-SVD to be too slow on problems of this size. For example, one sparse approximations of the signals, using a fast implementation of OMP [41], and one dictionary update approximately took 10 hours and this has to be repeated for a reasonable number of iterations, e.g. 1000 iterations!

V. Conclusions

We have presented a new algorithm for dictionary learning and have shown its advantages with different experiments and for different data sets. The proposed method is very flexible in using different constraints on the dictionaries. Because the problem of dictionary learning is considered in a more general form (bounded norm for dictionaries), better results were possible.

While some of the other methods are based on atom-wise dictionary update (K-SVD, MAP-DL with unit column-norm *a priori* information), the proposed method updates the whole dictionary at once. Although the computational complexity of each iteration of the given algorithm is roughly cubic, we found that the algorithm is much faster for large scale problems than, for example, K-SVD (which has a higher order of complexity).

The given method solves the dictionary learning problem in a unified framework. This unified framework provides extra flexibility to update the coefficients and the dictionary in a more efficient way. Furthermore, we showed the convergence of the method to a set of fixed points in this framework.

Finally we have shown that the constrained Frobenius-norm can increase the performance of dictionary learning by increasing the possible solution set. Audio coding with the learnt dictionary showed a superior rate-distortion performance over traditional orthogonal transform coding and overcomplete sparse coding with an oversampled DCT.

Appendix A
Matrix Form of the Majorizing Function

We can use the Taylor series to majorize the quadratic term of the objective function which has a bounded curvature. The

Taylor series in matrix form [42, Appendix D 1.7] is given by,

$$f(\mathbf{U}) = f(\mathbf{V}) + \overrightarrow{df}^{\mathbf{U-V}}(\mathbf{V}) + \frac{1}{2!}\overrightarrow{df}^{2\,\mathbf{U-V}}(\mathbf{V}) + o(||\mathbf{U}||^3) \quad (39)$$

where $\overrightarrow{df}^{\mathbf{U-V}}(\mathbf{V})$ and $\frac{1}{2!}\overrightarrow{df}^{2\,\mathbf{U-V}}(\mathbf{V})$ are the directional first and second derivatives of f at $\mathbf{V}$ in the $\mathbf{U}-\mathbf{V}$ direction. The directional derivatives are defined by,

$$\overrightarrow{df}^{\mathbf{Y}}(\mathbf{X}) = \{\frac{d}{dt}f(\mathbf{X}+t\mathbf{Y})\}_{t=0}, \quad \overrightarrow{df}^{2\,\mathbf{Y}}(\mathbf{X}) = \overrightarrow{df}^{\mathbf{Y}}(\overrightarrow{df}^{\mathbf{Y}}(\mathbf{X})).$$

For a bounded curvature objective function we have,

$$f(\mathbf{U}) \leq f(\mathbf{V}) + \overrightarrow{df}^{\mathbf{U-V}}(\mathbf{V}) + \frac{1}{2}tr\{(\mathbf{U}-\mathbf{V})^T\Pi(\mathbf{U}-\mathbf{V})\}, \quad (40)$$

where $\Upsilon = \Pi - \overrightarrow{df}^{2\,\mathbf{U-V}}(\mathbf{V})$ is positive definite ($\Upsilon \succ 0$).

Appendix B
Convergence Study of the Algorithm

In the first step of analyzing an iterative algorithm, we need to show the boundedness of the solutions (or the stability of the algorithm). The stability of the algorithms, in which a positive objective is reduced in each iteration, is guaranteed using Lyapunov's second theorem. For example the stability of the MAP-DL is guaranteed when a *suitable* step size is chosen (to the authors knowledge, no analytical study has been done on how to choose this step size). The convergence of the alternating (gradient) projection based methods essentially depends on the admissible sets (and the gradient step size). In the dictionary learning problem with the admissible sets given by [13] [11], the convergence of the algorithm is not guaranteed. In K-SVD, one needs to find the sparse approximations based on the ℓ_0 sparsity measure for which no efficient algorithm exists so that the stability analysis is challenging. In practice we observed that in MOD and K-SVD, when the solution sequence enters a neighborhood of a local minimum, the objective increases in some iterations. Therefore, it does not converge monotonically to the solution.

The next step is to show the convergence of the algorithm to a fixed point or a set of fixed points. The authors in [23] referred to the convergence of the gradient flow method to show the convergence of the MAP-DL. Although this statement is completely correct, it requires the use of an arbitrary small step size which is practically impossible.

The stability of dictionary learning based on the majorization method has already been proven by the fact that we reduce the objective in each step. Here, we show the convergence to a set of fixed points. Our dictionary learning framework can be viewed as a generalized block-relaxed minimization scheme applied to an augmented objective function. Specifically, we combine two majorizing objectives, (15) and (18),

$$\psi(\mathbf{D},\mathbf{X},\mathbf{D}^\ddagger,\mathbf{X}^\ddagger) = \phi(\mathbf{D},\mathbf{X}) + c_D||\mathbf{D}-\mathbf{D}^\ddagger||_F^2 + c_X||\mathbf{X}-\mathbf{X}^\ddagger||_F^2 - ||\mathbf{D}\mathbf{X}-\mathbf{D}^\ddagger\mathbf{X}^\ddagger||_F^2 \quad (41)$$

where $\mathbf{X}^\ddagger$ and $\mathbf{D}^\ddagger$ are two auxiliary parameters corresponding to $\mathbf{X}$ and $\mathbf{D}$ respectively. c_D and c_X have been chosen to be larger than the spectral norms of $\mathbf{X}^{\ddagger T}\mathbf{X}^\ddagger$ and $\mathbf{D}^{\ddagger T}\mathbf{D}^\ddagger$ respectively. This augmented objective function *does not* majorize the joint objective, however when $(\mathbf{D},\mathbf{D}^\ddagger|_{\mathbf{D}^\ddagger=\mathbf{D}})$ or $(\mathbf{X},\mathbf{X}^\ddagger|_{\mathbf{X}^\ddagger=\mathbf{X}})$ are fixed, (41) majorizes the original joint objective based on the other pair of parameters. When the optimization method is viewed in the block relaxation framework, the optimum of $\mathbf{X}^\ddagger$ or $\mathbf{D}^\ddagger$ is easily found by $\mathbf{X}$ or $\mathbf{D}$ respectively. This corresponds to the parameter update in the standard majorization method [29]. Therefore any sequence of updates is acceptable, given each update of $\mathbf{D}$ (or $\mathbf{X}$) is followed by an update based on $\mathbf{D}^\ddagger$ (or $\mathbf{X}^\ddagger$) respectively.

Such a block-relaxed sequential constrained minimization is not in general guaranteed to converge (see [24] for some counter examples). To study the convergence of our algorithm, we need to do a little more work. In the next subsection, we introduce some theoretical analysis of the generalized block relaxation method. We then analyze the proposed algorithm for dictionary learning, based on the given theoretical analysis.

A. Generalized Block relaxed iterative mappings and their convergence

Let $\eta(\omega) : \Omega \rightarrow \mathbb{R}$ be the multi-parameter objective function which we want to minimize. Let Υ be the set of admissible parameters. The parameter ω is defined as the concatenation of the blocks of parameters $\{\omega \in \Upsilon : \omega = (\omega_1,\omega_2,...,\omega_p), \omega_i \in \Omega_i\}$ where $\Omega = \Omega_1 \times \Omega_2 \times ... \times \Omega_p$. In dictionary learning based on block relaxation, $p = 2$, $\omega_1 = \mathbf{X}$ and $\omega_2 = \mathbf{D}$. In generalized block-relaxed dictionary learning, $p = 4$ as we have two more auxiliary parameters $\mathbf{X}^\ddagger$ and $\mathbf{D}^\ddagger$.

We now need to introduce point to set maps,

***Definition* B.1** (Point to set map)**.** Let Υ be an arbitrary set and let Γ be the set of all subsets of Υ. A map $\Delta : \Upsilon \rightarrow \Gamma$ is a point to set map (see for example [43]).

In the block relaxation technique a set of point to set maps $\Delta_i : \Upsilon \rightarrow \Gamma$ are defined as $\Delta_i(\widehat{\omega}) = \{\omega \in \Upsilon : \forall j \neq i \ \omega_j = \widehat{\omega}_j\}$ where $\widehat{\omega} = (\widehat{\omega}_1,\widehat{\omega}_2,...,\widehat{\omega}_p)$ is the current value of the parameters. These point to set maps keep all the blocks of parameters fixed apart from the i^{th} block.

By starting from $\omega^{[0]}$, the set of possible solutions Λ in the minimization problem is defined as, $\Lambda = \{\omega \in \Upsilon : \eta(\omega) \leq \eta(\omega^{[0]})\}$. For any $\omega \in \Lambda$ in each block update we minimize the objective for the selected parameters. This gives us the following updating operator:

$$U_i : \Lambda \rightarrow \{u \in \Delta_i(\widehat{\omega}) : \eta(u) \leq \eta(t), \forall t \in \Delta_i(\widehat{\omega})\} \quad (42)$$

In general this updating operator is a point to set map and we can choose an update parameter within the resulting set. In our case, the objective function always has a unique minimizer and the updating operators are point-to-point mappings. To use a set of updating operators, we also need to have an operator selector.

***Definition* B.2** (Operator selector)**.** $s(k) : \mathbb{N} \rightarrow \mathscr{P}$ which $\mathscr{P} = \{i : 1 \leq i \leq p\}$

This operator can choose the updating operator by sequentially selecting (circular) or free steering through the available operators. By using the updating operators defined in (42) and

an update selector $s(k)$, we can summarize the (generalized) block relaxed minimization by the following algorithm.

***Algorithm* B.1.** *Let $\omega^{[0]}$ be a given starting point, then $\{\omega^{[k]}\}_{k\in\mathbb{N}}$ is the sequence of updates given by $\omega^{[k+1]} \in U_{s(k)}\{\omega^{[k]}\}$ and stop when $\forall i \in \mathscr{P} : \widehat{\omega} = U_i\{\widehat{\omega}\}$*

When the updating operator is injective, $\omega^{[k+1]} = U_{s(k)}\{\omega^{[k]}\}$, to analyze the sequence generated by Algorithm B.1, we need to introduce some characteristics of the infinite series.

***Definition* B.3** (Asymptotically regularity)**.** A sequence $\{\alpha^{[n]}\}_{n\in\mathbb{N}}$ is asymptotically regular if $||\alpha^{[n+1]} - \alpha^{[n]}|| \to 0$, when $n \to \infty$.

$||\ .\ ||$ is a norm defined in the solution space. An operator is called asymptotically regular when the series generated by the sequential use of that operator is asymptotically regular.

***Definition* B.4** (Essentially periodic)**.** An infinite sequence $\{\alpha^{[n]}\}_{n\in\mathbb{N}}$ drawn from a finite alphabet $\mathscr{P} = \{\mathscr{A}_i : 1 \leq i \leq p\}$ is essentially periodic, with a period $m \in \mathbb{N}, m \geq p$ when $\forall j \in \mathbb{N}, \forall \mathscr{A}_i \in \mathscr{P}, \exists n \in [jm+1, (j+1)m]$ and $\alpha^{[n]} = \mathscr{A}_i$.

The sequence of $\{\omega^{[k]}\}$ of the Algorithm B.1 is asymptotically regular when Δ_i and η satisfy the following hypotheses [44],

***Hypotheses* B.1.** For all $i \in \mathscr{P}$ and $\eta : \Upsilon \to \mathbb{R}$,

- $\forall \omega : \omega \in \Delta_i(\omega)$
- Δ_i is continuous on Υ
- $\forall \omega \in \Upsilon$, η has a unique minimizer over $\Delta_i(\omega)$
- $\exists \omega^{[0]} \in \Upsilon$ such that Λ is a compact subset.

We now study the accumulation points of Algorithm B.1, when the Hypotheses B.1 are satisfied. From basic mathematical analysis, we know that any bounded sequence has at least one accumulation point (Bolzano-Weierstrass Theorem [45, Theorem 4.1]). As Λ is closed, the accumulation points of $\{\omega^{[n]}\}$ are in Λ.

***Theorem* B.1.** *[44, Theorem 15] Let the update selector, $s(k)$, be essentially periodic and Δ_i and η satisfy Hypotheses B.1. Every accumulation point ω^* of $\{\omega^{[n]}\}$, generated by Algorithm B.1, satisfies $\omega^* = U_i\{\omega^*\}$ for any $i \in \mathscr{P}$*

The set of accumulation points T belongs to a level set of η. If η is continuous, T is closed and as Λ is bounded and $T \subseteq \Lambda$, T is bounded. Therefore T is compact.

***Proposition* B.1.** [29, Proposition 10.3.1] If a bounded sequence $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ is asymptotically regular, then its set of accumulation points is connected. If this set is finite, then it reduces to a single point.

In a normed space, the following lemma guarantees that the sequence $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ generated by Algorithm B.1 will stay arbitrarily close to the accumulation points, when $n > N$ for some N.

***Lemma* B.1.** *Let $\{\omega^{[n]}\}_{n\in\mathbb{N}}$ be a bounded asymptotically regular sequence and T be the set of its accumulation points then, $\forall \epsilon > 0, \exists N \in \mathbb{N}, for\ n > N, \exists t \in T, ||\omega^{[n]} - t|| < \epsilon$*

Proof: Let S be an ϵ-neighborhood of T and S_c be its complement in the admissible set. As the admissible set is compact, S_c is also compact. Because S is a neighborhood of T there is no accumulation point t in S_c. If $\{\omega^{[n]}\}$ has infinitely many points in S_c, then it has a converging subsequence and at least one accumulation point in S_c . This contradicts the fact that there is no accumulation point in S_c. Therefore $\exists N : \omega^{[n]} \in S,\ \forall n > N$. On the other hand ϵ-neighborhood implies that for all $n > N$, $\exists t \in T$: $||\omega^{[n]} - t|| < \epsilon$. ■

In the next subsection we show asymptotic regularity of the generalized block relaxation method for dictionary learning. This is followed by showing the convergence of the proposed method to a set of fixed points.

B. Convergence study of the generalized block-relaxed dictionary learning

In dictionary learning, there are two parameters, coefficient matrix and dictionary. In generalized block-relaxed dictionary learning (41), we have four parameters. We mentioned that the augmented function (41) majorizes (6) only when one pair of parameter blocks ($(\mathbf{D}, \mathbf{D}^\ddagger|_{\mathbf{D}^\ddagger=\mathbf{D}})$ or $(\mathbf{X}, \mathbf{X}^\ddagger|_{\mathbf{X}^\ddagger=\mathbf{X}})$) is fixed. Therefore $\Delta_{\mathscr{X}} : \mathscr{X} \in \{\mathbf{D},\ \mathbf{X},\ \mathbf{D}^\ddagger,\ \mathbf{X}^\ddagger\}$ are the point to set maps which fix all parameters but $\mathscr{X}$ (from now on we use this indexing for the point to set maps).

***Proposition* B.2.** The generalized block-relaxed minimization of (41) is asymptotically regular when the updates of $\mathbf{D}$ and $\mathbf{X}$ are followed by updating of $\mathbf{D}^\ddagger$ and $\mathbf{X}^\ddagger$ respectively.

Proof: To show the asymptotic regularity we show that all the hypotheses in Hypotheses B.1 are satisfied. $\Delta_{\mathscr{X}} : \mathscr{X} \in \{\mathbf{D},\ \mathbf{X},\ \mathbf{D}^\ddagger,\ \mathbf{X}^\ddagger\}$ are self contained, i.e. $\widehat{\mathscr{X}} \in \Delta_{\mathscr{X}}\{\widehat{\mathscr{X}}\}$, and continuous. Therefore they satisfy the first two hypotheses. The minimum of (41) based on each parameter is unique (the sparse approximation minimum is reached using soft shrinkage (17) over $\mathbf{A}$ and the dictionary update is reached by one of the operators introduced in (24), (31) or (38) over $\mathbf{B}$). (41) is strictly convex based on $\mathbf{X}^\ddagger$ or $\mathbf{D}^\ddagger$ when all other parameters are fixed. Therefore minimization based on $\mathbf{D}^\ddagger$ or $\mathbf{X}^\ddagger$ has a unique solution. Surrogate objective function (41) is a continuous function. When a mapping is continuous, its epigraph Λ is a closed set [38, Theorem7.1]. As the admissible set is a closed set, the intersection of Λ and this set, which is the possible solution set, is closed. On the other hand there is no infinitely large point in Λ (maximum value of $||\mathbf{D}||_F$ and $J_{1,1}(\mathbf{X})$ are bounded based on the dictionary constraints and $\phi(\mathbf{D}^{[0]}, \mathbf{X}^{[0]})/\lambda$ respectively). In an Euclidean space boundedness and closedness are sufficient for a set to be compact. Therefore the hypothesis is satisfied and the sequence of $(\mathbf{D}, \mathbf{X}, \mathbf{D}^\ddagger, \mathbf{X}^\ddagger)^{[i]} : i \in \mathbb{N}$ is asymptotically regular [44]. ■

Finally we present a Proposition which shows the convergence of the proposed algorithm.

***Proposition* B.3.** Generalized block-relaxed dictionary learning converges to a single fixed point $(\mathbf{D}^*, \mathbf{X}^*)$ or gets arbitrary close to a continuum of accumulation points, where each accumulation point satisfies:

- $\psi(\mathbf{D}^*, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) \leq \psi(\mathbf{D}^*, \mathbf{X}, \mathbf{D}^*, \mathbf{X}^*) : \forall \mathbf{X}$

- $\psi(\mathbf{D}^*, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) \leq \psi(\mathbf{D}, \mathbf{X}^*, \mathbf{D}^*, \mathbf{X}^*) : \forall \mathbf{D} \in \mathscr{D}$

Proof: Due to Proposition B.2, the sequence generated by generalized block-relaxed dictionary learning is asymptotically regular. Due to Theorem B.1 and Lemma B.1, the algorithm converges either to a fixed point or gets arbitrary close to a continuum of accumulation points. Because any accumulation point of the algorithm is a fixed point for all $U_i : \forall i \in \mathscr{P}$ [44, Theorem 15], $\mathbf{X}^*$ is the best coefficient matrix using dictionary $\mathbf{D}^*$ and $\mathbf{D}^*$ is the best admissible dictionary, using $\mathbf{X}^*$ as the sparse representation. ■

REFERENCES

[1] I. Daubechies, *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.

[2] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[3] V. Goyal, J. Kovacevic, and J. Kelner, "Quantized frame expansions with erasures," *Applied and Computational Harmonic Analysis*, vol. 10, pp. 203–233, 2001.

[4] http://www.compressedsensing.com.

[5] G. Davis, "Adaptive nonlinear approximations," Ph.D. dissertation, New York University, 1994.

[6] S. Mallat and Z. Zhang, "Matching pursuits with time frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[8] S. Sardy, A. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.

[9] M. Elad and A. Bruckstein, "A generalized uncertainty principle and sparse representations in pairs of bases," *IEEE Trans. Information Theory*, vol. 48, no. 9, p. 25582567, 2002.

[10] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, 2003.

[11] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[12] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.

[13] K. Engan, S. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[14] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[15] M. Aharon, E. Elad, and A. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[16] K. Engan, K. Skretting, and J. Husoy, "Family of iterative ls-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digital Signal Processing*, vol. 17, no. 1, pp. 32–49, 2007.

[17] E. Smith and M. S. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Computation*, vol. 17, no. 1, pp. 19–45, 2005.

[18] M. Plumbley, "Dictionary learning for L1-exact sparse coding," in *International Conference on Independent Component Analysis and Signal Separation, ICA*, 2007.

[19] R. Gribonval and K. Schnass, "Some recovery conditions for basis learning by L1-minimization," in *International Symposium on Communications, Control and Signal Processing, ISCCSP*, 2008.

[20] S. Cotter, B. Rao, K. Engan, and K. Kreutz Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[21] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Trans. on Image Process*, vol. 16, no. 12, pp. 2980–2991, 2007.

[22] E. Cands, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," Caltech University, Tech. Rep., 2007.

[23] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, and T. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.

[24] J. Leeuw, "Block-relaxation algorithms in statistics," in Information Systems and Data Analysis, ed. H.H. Bock, W. Lenski and M. M. Richter, Berlin: Springer-Verlag, pp. 308-325, 1994.

[25] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annual of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[26] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math*, vol. 57, pp. 1413–1541, 2004.

[27] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[28] Z. Zhang, J. Kwok, and D. Yeung, "Surrogate maximization/minimization algorithms and extensions," *Machine Learning*, vol. 69, no. 1, pp. 1–33, 2007.

[29] K. Lange, *Optimization*. Springer-Verlag, 2004.

[30] A. Lyapunov, *Stability of motion*. Academic Press, 1966.

[31] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[32] M. Fornasier and H. Rauhut, "Iterative thresholding algorithms," to appear in Applied and Computational Harmonic Analysis, 2007.

[33] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 346–367, 2007.

[34] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[35] L. Landweber, "An iterative formula for fredholm integral equations of the first kind," *American Journal of Mathematics*, vol. 73, pp. 615–624, 1951.

[36] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide http://www.netlib.org/lapack/lug/lapack_lug.html, Third Edition*. Society for Industrial and Applied Mathematics (SIAM), 1999.

[37] O. Divorra Escoda, L. Granai, and P. Vandergheynst, "On the use of a priori information for sparse signal approximations," *IEEE Trans. on Signal Processing*, vol. 54, no. 9, pp. 3468–3482, 2006.

[38] R. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[39] M. Fornasier and H. Rauhut, "Recovery algorithms for vector valued data with joint sparsity constraints," to appear in SIAM Journal of Numerical Analysis, 2007.

[40] L. Daudet and B. Torresani, "Hybrid representations for audiophonic signal encoding," *Signal Processing, special issue on Coding Beyond Standards*, vol. 82, no. 11, pp. 1595–1617, 2002.

[41] "Sparsify 0.2," the University of Edinburgh, 2007.

[42] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2005 (v2007.09.17).

[43] W. Zangwill, *Nonlinear Programming: A Unified Approach*. Printice-Hall, 1969.

[44] J. Fiorot and P. Huard, "Composition and union of general algorithms of optimization," *Mathematical Programming Study*, vol. 10, pp. 69–85, 1979.

[45] B. Palka, *An Introduction to Complex Function Theory*. Springer, 1991.

Mehrdad Yaghoobi (S'98-M'09) received the BSc. and MSc. in electrical and biomedical engineering in 1999 and 2002 from the University of Tehran and Sharif University of Technology, respectively. He spent one year as a research assistant in the AICTC, Sharif University of Technology before starting his PhD at Queen Mary University of London, in December 2005. He moved to the University of Edinburgh to accompany his supervisor in April 2006. He is now persuing the PhD degree in the Institute for Digital Communications (IDCom) at the University of Edinburgh. His current research interests include sparse approximation, dictionary selection, compressed sensing and audio modelling/coding.

He is recipient of EPSRC studentship and the University of Edinburgh international tuition fee waiver.

Thomas Blumensath (S'02-M'06) received the BSc hons. degree in music technology from Derby University, Derby, U.K., in 2002 and his Ph.D. degree in electronic engineering from Queen Mary, University of London, U.K., in 2006. He is currently a postdoctoral research fellow in the Institute for Digital Communication at the University of Edinburgh. His research interests include mathematical and statistical methods in signal processing with a focus on sparse signal models and their application.

Mike E. Davies (M00) received the B.A. (Hons.) degree in engineering from Cambridge University, Cambridge, U.K., in 1989 and the Ph.D. degree in nonlinear dynamics and signal processing from University College London, London (UCL), U.K., in 1993. Mike Davies was awarded a Royal Society Research Fellowship in 1993 and was an Associate Editor for IEEE Transactions in Speech, Language and Audio Processing, 2003-2007.

He currently holds the Jeffrey Collins SHEFC funded chair in Signal and Image Processing at the University of Edinburgh. His current research interests include: sparse approximation, compressed sensing and their applications.

IEEE TRANSACTION ON SIGNAL PROCESSING, VOL. X, NO. X, JANUARY 20XX 1

Parametric Dictionary Design for Sparse Coding

Mehrdad Yaghoobi, *Member, IEEE,* and Laurent Daudet, *Member, IEEE* and Mike E. Davies, *Member, IEEE*

***Abstract*—This paper introduces a new dictionary design method for sparse coding of a class of signals. It has been shown that one can sparsely approximate some natural signals using an overcomplete set of parametric functions, e.g. [1], [2]. A problem in using these parametric dictionaries is how to choose the parameters. In practice these parameters have been chosen by an expert or through a set of experiments. In the sparse approximation context, it has been shown that an incoherent dictionary is appropriate for the sparse approximation methods. In this paper we first characterize the dictionary design problem, subject to a constraint on the dictionary. Then we briefly explain that equiangular tight frames have minimum coherence. The complexity of the problem does not allow it to be solved exactly. We introduce a practical method to approximately solve it. Some experiments show the advantages one gets by using these dictionaries.**

***Index Terms*—Sparse Approximation, Dictionary Design, Incoherent Dictionary, Parametric Dictionary, Gammatone Filter Banks, Exact Sparse Recovery.**

I. Introduction

SPARSE modeling of signals has recently received much attention as it has shown promising results in different applications. It has been used for coding, source separation, feature extraction and compressive sampling. A basic assumption to apply this model is that the given class of signals can be sparsely represented or approximated in an underdetermined generative model. Often, a linear model has been used as the generative model. In this framework, one can use a matrix $\mathbf{D}_{d\times N} \in \mathbb{R}^{d\times N} : d < N$, called dictionary, to represent the signal approximately using $\mathbf{y} \approx \mathbf{D}\mathbf{x}$.

Sparse approximation and sparse representation methods have been studied theoretically and practically [3]. Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the given signal and the coefficient vector respectively. A sparse approximation would be,

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} ||\mathbf{x}||_0 \text{ s.t. } ||\mathbf{y} - \mathbf{D}\mathbf{x}||^2 \leq \xi,$$

where $||.||_0$ is the sparsity measure that counts the number of the non-zero coefficients and ξ is a small positive constant. This problem in general, like the sparse representation problem ($\xi = 0$), is an NP-hard problem [4] and can not be solved in a reasonable time. Numerous algorithms have been proposed to find an approximate solution. These algorithms are classified as greedy methods, like Matching Pursuit (MP) [5] and its derivations [6], and relaxation methods, like Basis Pursuit Denoising (BPDN) [7] and IRLS-type algorithms [8], [9]. The sparsity of the representation can be increased using an appropriate dictionary for the given class of signals. The common methods for dictionary selection are to concatenate orthogonal bases, see for example [10] and [11] for the possible advantages of using such a dictionary in theory and practice, or to use a tight frame [12]. These dictionaries can be improved using dictionary learning methods [13]–[16]. These methods adapt an initial dictionary to a set of training samples. Therefore the aim is to *learn* a dictionary for which an input signal, taken from a given class of signals, has a sparse approximation.

There is another dictionary selection method, which is called dictionary *design*. Different methods exist to design a suitable $\mathbf{D}$ for a set of natural signals. One method is based on a generative model of the signals. If these signals are to be received by the human sensory system, a more effective method to design $\mathbf{D}$ is to use a human perception model [1], [2]. In fact, the stimuli responses generate elementary functions which are more related to the analysis dictionary [17]. These elementary functions have also been used for generating the synthesis dictionary $\mathbf{D}$. Here, we assume that the set of elementary functions can be described by using a set of parameters and a parametric function. For example, in the multiscale Gabor functions [5], the parameters are scale, time and frequency shifts and the parametric function is Gaussian. In general the parameters are in the continuous domain. To generate a dictionary based on these generative functions, we can sample these continuous parameters. The question is then how best to sample the parameters. Several researchers have introduced different methods to optimize the sampling process. In [18], a sampling scheme was introduced which finds an approximately tight frame, using 2D Gabor functions. Gammatone and Gammachirp filter banks have been shown to approximate the human auditory system. [19] presented two types of filters, which approximate the Gammatone filter banks, and allow a possible fast VLSI implementations. Alternatively, some researchers optimized the parameters based on the closeness to what is observed in the perceptual systems [20], [21], [22]. In practice, [23] showed that the optimal parameters, found by fitting to the human auditory system, do not match the parameters estimated from English speech signals.

When we use an approximate or a relaxed method to find a sparse approximation, having an exact generative model does not guarantee that we find the best sparse approximation. An important parameter of a dictionary, for successful sparse recovery, is its coherence μ [24]. The coherence is defined as the absolute value of the largest inner-product of two

Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This research was partially supported by EPSRC grant D000246/1 and EU FP7, FET-Open grant number 225913. M. Yaghoobi and M. Davies are with the Institute for Digital Communication and with the Joint Research Institute for Signal and Image Processing, Edinburgh University, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, UK (e-mail: yaghoobi@ieee.org, mike.davies@ed.ac.uk).

L. Daudet is with UPMC Univ. Paris 06, LAM / Institut Jean le Rond d'Alembert (UMR 7190), Paris, France (email: daudet@lam.jussieu.fr).

distinct atoms and it has been shown that when μ is smaller than a certain value MP and BPDN can recover the sparse representation of the input signal [10], [25], [26]. It has also been shown that the coherence upper-bounds the residual error decay in MP [27] and OMP [25]. Therefore a dictionary with small μ is desirable for sparse coding. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of the dictionary. The coherence of $\mathbf{D}$ is the maximum absolute value of the off-diagonal elements of $\mathbf{G}$, whenever the columns of the dictionary are normalized. For such $\mathbf{D}$ if the magnitude of all off-diagonal elements of $\mathbf{G}$ are equal, $\mathbf{D}$ has minimum coherence [28]. This normalized dictionary is called an Equiangular Tight Frame (ETF) [29]. Although this type of frame has various nice properties, we here consider its advantages in exact atom recovery [25] and the residual error decay rate [27]. Unfortunately ETF's do not exist for any arbitrary selection of d and N [29]. Therefore a dictionary design aim can be to find the nearest admissible solution. On the other hand, natural signals do not generally have sparse approximations using an ETF. Therefore, the dictionary design problem can be to find a parametric dictionary whose Gram matrix is close to the Gram matrix of an ETF. This way, domain knowledge is incorporated into the parametric functions used, while the optimization aims at improving the ability of algorithms to find sparse approximations. The given class of signals has a sparse approximation using the proposed dictionary. That is because it is generated by sampling the parameters of generative functions fitted to the signal, whilst the dictionary has nice properties that allow exact atom recovery, because it is close to being an ETF. In practice we show that the designed dictionary indeed gives advantages over the standard dictionary, in terms of efficient sparse approximation. Another advantage of the parametric dictionary is that sparse approximation methods only need to store the parameters, instead of the full dictionary, which offers a huge reduction in memory requirement (the size of the parameter matrix is much smaller than the size of the corresponding dictionary). Sometimes this type of parametric dictionary can furthermore be multiplied to the coefficient vectors faster than direct matrix-vector multiplication. It then also speeds up most of the currently available sparse coding methods.

The parametric dictionary design, like other dictionary design methods, has some disadvantages. The main disadvantage is that it does not explicitly depend on a given class of signals, but instead on a class of parametric dictionaries. As an example, if the actual data often lies in a subspace of the signal space, the optimal dictionary[1] would have more atoms in that subspace. This might contradict with the minimum coherence constraint. It is hoped that this can be prevented by appropriate choice of the parametric family of functions and the initialization of the algorithm. Another difficulty in the given problem is that the current algorithm stores the Gram matrix explicitly. The current method is thus not tractable for very large dictionaries.

It deserves to be mentioned that there is another way to use parametric dictionaries. In [30], [31] some methods are proposed to sparsely approximate signals using continuous parameter parametric dictionaries. The convergence rate of MP algorithm with this setting is also studied in [31]. In contrast the designed dictionary, using parametric dictionary design, is discretized and can be used by the conventional sparse coding methods.

[1]The optimal dictionary is that by which the given class of signals has the sparsest approximation.

A. Contributions of the paper

In this paper we introduce a new framework for dictionary design. To the authors knowledge, this formulation has not been considered previously. This formulation can be used to design a dictionary when dictionary learning is not possible, or is computationally intractable. We show how we can find an approximate solution using an alternating minimization type method.

The parametric dictionary is represented using a small number of parameters (often less than 5). Therefore we do not need to store the dictionary explicitly. This can save a considerable amount of memory when using sparse approximation algorithms.

Finally we show experimentally that there are sparse approximation benefits in using such a parametric dictionary for audio coding.

B. Organization of the paper

In the next section we formulate the parametric dictionary design problem. We then present a practical algorithm to find an approximate solution. For a case study we present the parametric dictionary formulation and the update formula derivation. Experiments, in the simulation subsection, show the advantages of the proposed dictionary design. The stability of the algorithm is analyzed after its introduction in Section III, while the convergence of the proposed algorithm is shown in Appendix A.

C. Notation

In this paper we use small and capital bold face characters to indicate vectors and matrices respectively. All the parameters have real values, even though we do not state this explicitly each time. The matrix and vector norm spaces that we use in this paper are defined over the real fields with ℓ_2 and $||.||_F$, which is the Frobenius norm, as the corresponding norms respectively.

The tensor product used in this paper is for the multiplication of two three-dimensional arrays. This multiplication uses the first two indices to make a simple matrix-matrix product and the third parameter as the indices of these products. In other words, the third parameter specifies two matices from the three dimensional tensors and simplifies the tensor product to matrix-matrix multiplications. The number of these multiplications is the size of third index.

The terms "ETF" and "Grassmannian Frame" have been used interchangingly for the same concept [32], [28], [29]. In this paper we prefer to use ETF, which is more comprehensive.

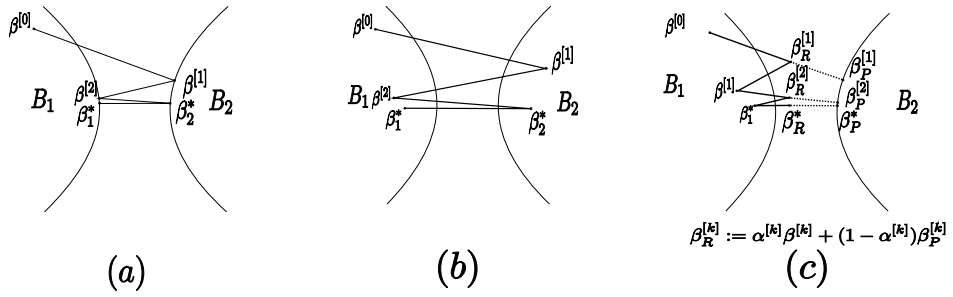

Fig. 1. Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method.

II. Parametric Dictionary Design: Formulation

In this section we formulate the problem of optimizing $\mathbf{D}$ to be close to an ETF. Let $\mathbf{D}_\Gamma \in \mathscr{D}$ be a parametric dictionary. Γ is the parameter matrix, with γ_i as its i^{th} column and $\mathscr{D}$ is the set of admissible parametric dictionaries. Each column of $\mathbf{D}_\Gamma$, $\mathbf{d}_i$ (with the associated parameters γ_i), is called an atom. In this paper, by letting $\mathbf{D}_\Gamma$ be a matrix, we implicitly assume that the generative model is discrete. This model can be extended to a continuous model, which is out of our scope. To select a $\Gamma \in \Upsilon$, where Υ is an admissible parameter set, we need to introduce an objective. In section I we explained that for a better performance in sparse coding, we are interested to design a dictionary which is close to being an ETF. For a given normalized $\mathbf{D}$, the coherence of $\mathbf{D}$, $\mu_\mathbf{D}$, is defined by,

$$\mu_\mathbf{D} = \max_{i,j:j\neq i} \{|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|\}.$$

A column normalized dictionary $\mathbf{D}_G$ is called ETF, or Grassmannian frame [32], when there is a $\gamma \ : \ 0 < \gamma < \pi/2$, such that,

$$|\langle \mathbf{d}_i, \mathbf{d}_j\rangle| = \cos(\gamma) \ : \ \forall i,j \ i \neq j.$$

The authors in [32] showed that if there exists an ETF in $\mathscr{D}$, the set of d by N uniform frames[2], it is the solution of,

$$\arg\inf_{\mathbf{D}\in\mathscr{D}} \{\mu_\mathbf{D}\}.$$

The infimum has been used to guarantee that the problem has at least a solution, when $\mathscr{D}$ is not closed, which is in the closure of $\mathscr{D}$. To study the lower bound of μ_D, the existence of an ETF and its Gram matrix, [32] introduced the following Theorem.

Theorem 1: [32, Theorem 2.3] Let $\mathbf{D}$ be a uniform frame in $\mathbb{R}^{d\times N}$. Then,

$$\mu_\mathbf{D} \geq \mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{1}$$

Equality holds in (1) *if and only if* $\mathbf{D}$ is an ETF. Furthermore, equality in (1) can only hold if $N \leq \frac{d(d+1)}{2}$.

Let Θ_d^N be the set of Gram matrices of all $d \times N$ ETFs. If $\mathbf{G}_G \in \Theta_d^N$ then the diagonal elements and the absolute values of the off-diagonal elements of $\mathbf{G}_G$ are one and μ_G respectively. A nearness measure of $\mathbf{D} \in \mathbb{R}^{d\times N}$ to the set of ETFs can be defined as the minimum distance between the Gram matrix of $\mathbf{D}$ and $\mathbf{G}_G \in \Theta_d^N$ [28]. To optimize the distance of a dictionary to an ETF, we can solve,

$$\inf_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} ||\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_G||_\infty,$$

where the matrix operator $||.||_\infty$ is defined as the maximum absolute value of the elements of the matrix. Instead, we would like to use a different norm space which simplifies the problem[3]. An advantage of using ℓ_2 measure in the given problem is that it considers the errors of all elements (and not just the maximum absolute error). In this framework, when there is no ETF in $\mathscr{D}$, we find a dictionary that is close to be quasi-incoherent [25] [27]. Therefore we use the following formulation,

$$\inf_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} ||\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_G||_F^2, \tag{2}$$

where $||.||_F$ is the Frobenius norm. This is a non-convex optimization problem in general. It might have a set of solutions or it may not have any solution (e.g. Θ_d^N is empty as there do not always exist ETF's for the arbitrary N and d). One can extend Θ_d^N to a convex set Λ^N [28], which is non-empty for any N, by

$$\Lambda^N = \{\mathbf{G} \in \mathbb{R}^{N\times N} : \mathbf{G} = \mathbf{G}^T, \operatorname{diag}\mathbf{G} = 1, \max_{i\neq j} |g_{i,j}| \leq \mu_G\}.$$

Relaxing (2), by replacing Θ_d^N with Λ^N, gives the following optimization problem,

$$\inf_{\Gamma\in\Upsilon, \mathbf{G}\in\Lambda^N} ||\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}||_F^2 \tag{3}$$

An important difference between (2) and (3) is that the relaxed problem, by using non-empty admissible sets, is guaranteed to have at least one solution. In this work, it is assumed that Υ is closed, which allows us to use the "min" operator instead of "inf" in (3). We therefore use the relaxed formulation from now on. We show experimentally that the approximate solutions of (3), even though the Gram matrix of the dictionary might only be close to Λ^N, show good performances in sparse approximation.

[2]A frame with unit column norms.

[3]Although the matrix space with ℓ_∞ is a well defined Banach space, here, we use ℓ_2 norm Hilbert space to use easy formulation of the optimization process.

Algorithm 1 *Parametric Dictionary Design*

1: **initialization:** $k=1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1\leq i\leq K} : 0<\alpha_i\leq 1$
2: **while** $k \leq K$ **do**
3: $\quad \mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
4: $\quad \mathbf{G}_{P_{k+1}} = \min_{\mathbf{G}\in\Lambda^N} ||\mathbf{G}_{\Gamma_k} - \mathbf{G}||_F$
5: $\quad \mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$
6: $\quad \mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : ||\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}||_F < ||\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}||_F\}$
7: $\quad k = k+1$
8: **end while**

Algorithm 2 *Parameters Update*

1: **initialization:** $l=1$, $1\leq L$, $\Gamma_k^{[0]} = \Gamma_k$, $\epsilon \in \mathbb{R}^+$, $\phi(\Gamma) = ||\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}||_F^2$
2: **for all** $l \leq L$ **do**
3: $\quad \Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$
4: $\quad l = l+1$
5: **end for**
6: $\Gamma_{k+1} = \Gamma_{k+1}^{[L]}$

In the next section we introduce a practical method to find an approximate solution to (3). Our approach has similarities with alternating minimization. This method is guaranteed not to increase the objective function in each step. Because the objective is non-negative, the algorithm is stable[4] due to Lyapunov's second theorem [33]. Also, one can show that the objective value converges. The stability of the algorithm and the convergence of the objective value do not prove the convergence of the algorithm. In Appendix A, it has been show that the algorithm converges to a set of accumulation points under mild conditions.

III. Parametric Dictionary Design: A Practical Algorithm

A standard method to solve (3) is alternating projection, see for example [34], [28] and references therein. In this method we alternatingly project the current solution onto the admissible sets, see Fig.1.a. When the admissible sets are convex, the algorithm converges[5] to a solution in $\mathscr{D}\cap\Lambda^N$ or a pair of solutions in $\mathscr{D}$ and Λ^N, when $\mathscr{D}\cap\Lambda^N = \emptyset$, respectively. In the following, we derive a formulation for the projection onto Λ^N, but there is no easy formulation for the projection onto the set of admissible dictionaries, in general. Therefore we choose a different method which has similarities with alternating minimization [36] (or generalized alternating projection [37]), see Fig.1.b. In the alternating minimization framework, we choose the new solutions in $\mathscr{D}$ and Λ^N alternatingly such that the objective does not increase in each update and is thus stable. If the algorithm converges, the fixed point is either in $\mathscr{D}\cap\Lambda^N$, or is a pair of points in $\mathscr{D}$ and Λ^N respectively.

Although the proposed algorithm has similarities with alternating minimization, it does not follow its steps exactly. The difference is that in the stage in which we update the current solution with respect to Λ^N, we choose a point which is somewhere between the current solution and the projection onto Λ^N. Fig.1.c shows a schematic representation of the proposed method. The reason for this modification is that by projection onto Λ^N, the structure of the Gram matrix changes significantly so that the selection of a new point in $\mathscr{D}$ in the following step is very difficult. We can gradually select a closer point to the projected point on Λ^N, when the current $\mathbf{D}_\Gamma$ is close to Λ^N. In the other step, we update $\mathbf{D}$ such that it does not increase the objective in (3).

The parametric dictionary design is summarized in Algorithm 1. In line 4, the algorithm finds the projection onto Λ^N. In line 6, a point in $\mathscr{D}$ is selected which is closer to $\mathbf{G}_{R_{k+1}}$. In the following we show how we calculate the updates in lines 4 and 6.

A. Projection onto Λ^N:

In the objective function (3), $\mathbf{G}$ is a Hermitian matrix. By sign change of any related off-diagonal pair of elements , i.e. $g_{i,j}$ and $g_{j,i}$, we get a new $\widetilde{\mathbf{G}} \in \Lambda^N$. The closest $\mathbf{G}$ to $\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma$, in a Frobenius norm space, is the $\mathbf{G}$ with a similar sign pattern. We know that in a normed space, finding the nearest elements of a set to a point is a projection of that point onto the set. Because Λ^N is convex, the projection is unique. For a given $\mathbf{G}_\mathbf{D} = \mathbf{D}^T\mathbf{D} : \mathbf{D} \in \mathbb{R}^{d\times N}$, the projection of $\mathbf{G}_D$ onto Λ^N can be found by the following operator [28].

$$g_{P\,i,j} = \begin{cases} \text{sign}(g_{D\,i,j})\mu_G & i \neq j \\ 1 & otherwise\ , \end{cases} \tag{4}$$

where μ_G is as defined in (1). This operator can be used to find $\mathbf{G}_{P_{k+1}}$ in line 4 of Algorithm 1, by applying to $\mathbf{G}_{\Gamma_k}$.

B. Parameter update:

Let us assume $\mathbf{D}_\Gamma$ is a differentiable function on Υ and therefore (3) is a differentiable function on Υ. An easy way to find Γ_{k+1}, such that it satisfies line 6 of the Algorithm 1, is to use the gradient descent method. We rewrite (3) as a minimization problem based on Γ when $\mathbf{G}_{R_{k+1}}$ is fixed.

$$\min_{\Gamma\in\Upsilon} \phi(\Gamma)\ ,\ \phi(\Gamma) := ||\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}||_F^2 \tag{5}$$

The gradient of the objective function in (5) can be found by chain rule for the matrix functions [38, D.1.3].

$$\begin{aligned} \nabla_\Gamma \phi &= \nabla_\Gamma \mathbf{D}_\Gamma\ \nabla_{\mathbf{D}_\Gamma}\phi \\ &= 4\nabla_\Gamma \mathbf{D}_\Gamma\ (\mathbf{D}_\Gamma\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{D}_\Gamma\mathbf{G}_{R_{k+1}}) \end{aligned} \tag{6}$$

In this formulation, one still needs to calculate $\nabla_\Gamma\mathbf{D}_\Gamma$. In Appendix B, we derive this formulation for a special parametric dictionary. We iteratively use the gradient descent method to find a *local* minimum of the problem (5). Let $\Gamma_k^{[0]} = \Gamma_k$, the updating formula is as follows,

$$\Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon\nabla_\Gamma\phi\big|_{\Gamma_k^{[l]}}, \tag{7}$$

where ϵ is a small positive value. The parameter ϵ should be chosen such that the update reduces the objective function in

[4]Here stability means boundedness of the algorithm output.
[5]At least in finite dimensional spaces. There are counter-examples for the lack of convergences in the infinite dimension setting [35].

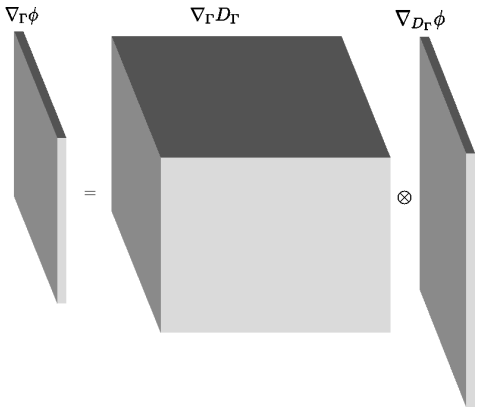

Fig. 2. The chain rule (6) in the tensor form.

(5) [39]. In this framework, $\Gamma_{k+1} = \lim_{l\to\infty} \Gamma_{k+1}^{[l]}$. In practice we stop after a given number of iterations or when $\nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$ becomes very small. Algorithm 2 summarizes this parameter update algorithm.

Because $\phi(\Gamma)$ is a continuous function, its epigraph [40], for an initial Γ_0[6], is closed. By choosing a bounded set of admissible parameters Υ, the epigraph is a compact set in Euclidean space. To show that the algorithm gets as close as possible to the set of limit points, we need to use the Bolzano-Weierstrass theorem.

Theorem 2: [41, 3.24] Every bounded infinite subset of $\mathbb{R}^N$ has at least one limit point in $\mathbb{R}^N$.

Therefore, when the set of admissible parameters is bounded and ϵ is selected such that moving in the gradient direction with this step size reduces the objective, this gradient descent algorithm has at least one limit point in the admissible set.

Remark 1: The function $\phi(\Gamma)$ is a lower bounded function. Hence, if we reduce ϕ in each iteration, due to Lyapunov's second theorem [33], the algorithm is stable.

Remark 2: Algorithm 1 is an iterative algorithm in which we also used another iterative method for the dictionary update in line 6. The stability and the convergence of the updates mentioned above were related to the inner loop in Algorithm 1. We deal with the convergence of Algorithm 1 in Appendix A.

Remark 3: We draw the readers attention to the formulation (6). The parameters $\nabla_\Gamma \mathbf{D}_\Gamma$, $\nabla_{D_\Gamma} \phi$ and $\nabla_\Gamma \phi$ are tensors of rank 3, 2 and 2 respectively. If $\Gamma \in \mathbb{R}^{p\times N}$ and $\mathbf{D} \in \mathbb{R}^{d\times N}$ then $\nabla_\Gamma \mathbf{D}_\Gamma \in \mathbb{R}^{p\times d\times N}$, $\nabla_{D_\Gamma} \phi \in \mathbb{R}^{d\times 1\times N}$ and $\nabla_\Gamma \phi \in \mathbb{R}^{p\times 1\times N}$. A graphical presentation of this formulation is presented in Fig. 2 . Furthermore, to use this directional update in (7), we need to map $\nabla_\Gamma \phi \in \mathbb{R}^{p\times 1\times N}$ into the appropriate matrix in $\mathbb{R}^{p\times N}$. It is easily done by changing the order of indices (1,2,3 to 1,3,2), following by cancelling the third dimension. Because the rank of $\nabla_\Gamma \phi$ is 2, this mapping is injective.

[6]Epigraph of $\phi(\Gamma) : \Upsilon \to \mathbb{R}$ for an initial Γ_0 is defined [40, 3.1.7] by: $\text{epi}\,\phi = \{\Gamma \ : \ \Gamma \in \Upsilon, \phi(\Gamma) \le \phi(\Gamma_0)\}$

IV. Case study

The problem we formulated in this paper is developed in a general form. To show the advantages of using parametric dictionary design, we choose a case study. In sparse audio processing, an important question is how to choose the dictionary [42], [43]. Different methods have been introduced to adapt the dictionary to better fit a set of training samples [44], [45], [46]. For example, some researchers used a class of parametric dictionaries based on Gammatone filter banks, which have been shown to have similarities with the human auditory system [23], [47]. We now show that the parametric dictionary design improves the performance of audio sparse approximation and exact recovery based around a Gammatone representation.

A. Gammatone parametric dictionary

The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t), \tag{8}$$

where $B = f_c/Q + b_{min}$, f_c is the center frequency and $n \in \mathbb{N}$, a, b, Q, b_{min} are some constants. The optimal parameter selection is not easy. One can select the parameters such that the generated atoms match the auditory impulse response. The auditory system has been optimized through evolution and may not be optimized for a practical application. Our goal is to optimally select these parameters so that sparse approximation methods can be used. Another difficulty in using the Gammatone filter banks as a dictionary is its large size. A moderate size dictionary can be designed by the proposed method.

The dictionary is generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift. In this paper, $\gamma = [t_c \ f_c \ n \ b]^T$ are the optimization parameters. The parameters t_c and f_c change the center of the atoms in the time-frequency plane. n and b control the rise time and the width of the atoms in the time domain, respectively. The parameter a is chosen to normalize the atom to unit length. Let $\{\gamma_i\}_{1\le i\le N}$ be a set of the parameters and $g_{\gamma_i}(t)$ be the atom generated using γ_i. The parameter matrix Γ and the parametric dictionary $\mathbf{D}_\Gamma$ are generated using γ_i and $g_{\gamma_i}(\lfloor t f_{samp} \rfloor)$ as the columns respectively, where f_{samp} is the sampling frequency.

The differentiability of $\mathbf{D}_\Gamma$ with respect to Γ makes the parameter update easier. In this paper we assume the parametric dictionary satisfies this constraint. Letting $n \in \mathbb{R}$, (8) becomes a generative function over a continuous domain Υ. This function is differentiable with respect to Γ. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, Υ is a compact set thus guaranteeing that the algorithm converges to a set of fixed points. A necessary modification in Algorithm 1 is to use a mapping to Υ, when at least one parameter goes out of Υ, and comparing to the previous solution (to make sure that we do not increase the objective by the parameter update). A simple mapping operator is the thresholding operator, where it chooses the closest admissible parameter.

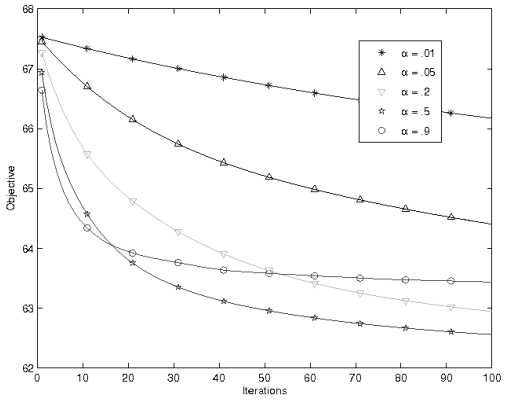

Fig. 3. The objective functions for different $\{\alpha_k\}_{\forall k,\alpha_k=\alpha}$, for a constant α.

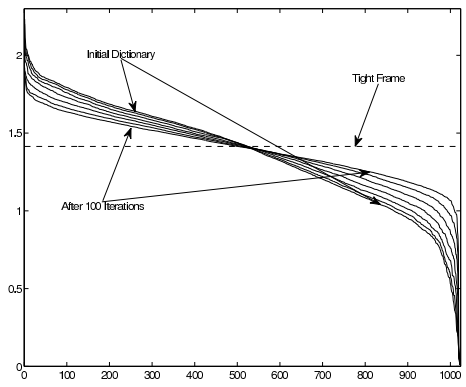

Fig. 4. Eigen values plot of the dictionary.

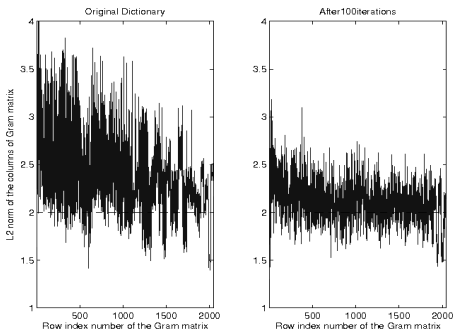

Fig. 5. The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries.

Although the computation of the gradient of a parametric dictionary generated using $g(t)$ is straightforward, we derive it in the Appendix B for completeness.

B. Simulations results

We study the proposed dictionary design method using the Gammatone dictionary discussed in IV. We first investigate the characteristics of the dictionaries throughout the design iterations. The stability of the algorithm is demonstrated by showing the reduction of the objective function. In the second part of this subsection, we compare the performance of the initial and the optimized dictionaries in terms of sparse approximation and exact sparse recovery. Gammatone type dictionaries have been proposed for sparse approximation of audio and we choose our examples accordingly. In all the simulations we choose two times overcomplete dictionaries and window size 1024.

1) Algorithm Evaluation: In this part, we evaluate the given algorithm in three different areas. In the first step we show that the algorithm reduces, (or at least keep the same) the objective (3) in each iteration. The parameter B, defined after (8), is the bandwidth of the audio filterbank at the center frequency f_c. We use the fixed values $n = 4$, $Q = 9.26449$, $b_{min} = 24.7$, as they have been suggested in [48] and [49], and $b = 0.65$. To generate the initial dictionary, we sample f_c and t_c. We use the method introduced in [50] to generate the filter bank. In this method an extra parameter δ, called step factor, is introduced to indicate the amount of frequency overlap. In this framework the k^{th} frequency center is calculate using the following formula,

$$f_c^k = -Qb_{min} + (f_s/2 + Qb_{min})e^{-k\delta/Q}. \qquad (9)$$

f_s is the maximum allowed frequency, which is half of the Nyquist frequency. In our simulations, we choose $\delta = 0.45$. We have chosen a similar method to sample t_c. This time sampling is linear, in contrast with the logarithmic sampling in (9). Let the peak of the envelope of the impulse response of the filter be at t_p and σ indicate the amount of time overlap. The l^{th} time center is found using,

$$t_c^l = t_p + \sigma(l-1)\ t_p\ .$$

σ is set to 0.75 in our simulations. We draw the readers attention to the point that t_c^l is implicitly a function of f_c^k. We therefore generate a set of $\{f_c^k\}_{k\in\mathscr{K}}$ and for each generated atom using f_c^k and $t_c = 0$, we make a set of time-shifted versions using $\{t_c^l\}_{l\in\mathscr{L}}$.

To generate a dictionary of $g_{\gamma_i}(t)$, we window it to a size equal to the signal length d and make it periodic such that one period is selected as an atom using the following formula,

$$\mathbf{d}_{\gamma_i,j} = \begin{cases} g_{\gamma_i}(j+d) & 1 \le j < j_{c_i} \\ g_{\gamma_i}(j) & j_{c_i} \le j \le d, \end{cases} \qquad (10)$$

where $j_{c_i} = \lfloor t_{c_i} f_{samp} \rfloor$. As the proposed algorithm is a relaxed version of the alternating minimization, the relaxation

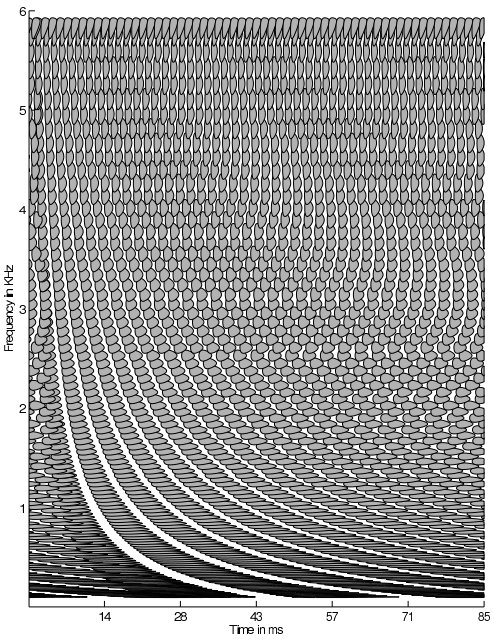

Fig. 6. Wigner-Ville contour plots of the original Gammatone atoms. The WV contour of each atom is calculated at 0.7 times it peak.

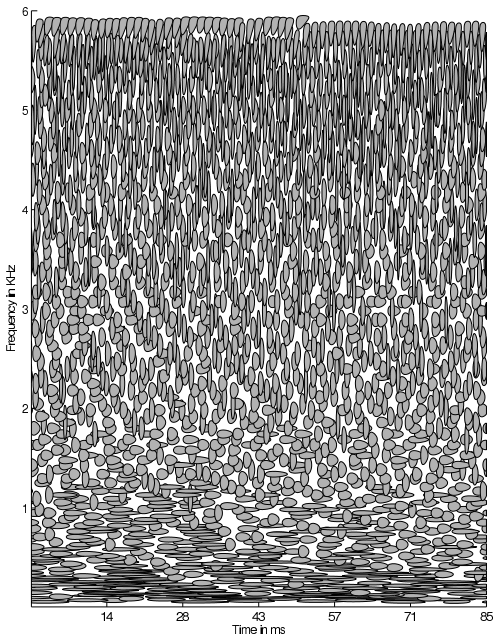

Fig. 7. Wigner-Ville contour plots of the learned Gammatone atoms. The contours are calculated similar to Fig. 6

parameters $\{\alpha_k\}$ should be selected. We choose a simple sequence of $\{\alpha_k\}$ using $\alpha_k = \alpha$ for all k and a fixed α in all simulations. A more complicated sequence might improve the performance of Algorithm 1. However we have not present this here. Here, we intend to show that the designed dictionary is superior to the initial dictionary in practice, even with a simple $\{\alpha_k\}$. In the first experiment we want to investigate the effect of α. We have plotted the objective function (3) using selected α's, in Fig. 3. As we expect, simulations show reduction of the proposed objectives in each iteration. It is also demonstrated that if α is small, the algorithm converges very slowly. Although using a large α is desireable for a fast convergence, the solution is not as good as the solution found by using a medium range α. For other simulations we use $\alpha = 0.5$ to find a good solution after an acceptable number of iterations.

The proposed algorithm searches for an equiangular *tight frame*. Therefore one way to show the performance of the proposed algorithm is to compare the singular values (SV) of the designed dictionary and the tight frame. A tight frame in $\mathbb{R}^{d\times N}$ has d non-zero SV equal to $\sqrt{N/d}$. We have plotted the sorted SV's of the dictionaries at selected iterations in Fig. 4. It can be seen that the SV's of the designed dictionary get closer to the SV's of the tight frame at each iteration.

Given that the algorithm is based on distances in the Gram matrix domain, another way to evaluate the algorithm is to show the Gram matrix of the dictionary. We have plotted the ℓ_2 norm of each row of the Gram matrix, for the window size 1024, in Fig. 5. The Gram matrix of the original dictionary and the designed dictionary, after 100 iterations, are shown in the left and right windows respectively. We have shown the ℓ_2 norm of a possible ETF with a dashed line as reference. It can be seen that the Gram matrix of the designed dictionary is closer to the desired Gram matrix. Another observation in Fig. 5 is that the atoms with high total cross-correlations, indicated by the peaks, are adapted.

This parametric dictionary is attempting to tile the time-frequency plane. An ETF is a frame having the minimum total correlation between atoms but it may not be localized in the time-frequency plane. A dictionary which is simultaneously ETF, or close to being an ETF, and localized in time and frequency, tiles time-frequency plane more uniformly. To demonstrate this, we choose the Wigner-Ville (WV) time-frequency representation of the atoms. We show the contour plot of the atoms in the time-frequency plane using a similar method used in [51]. Fig. 6 and 7 show the time-frequency planes found for the original and designed atoms, respectively. Although the algorithm attempts to minimize μ by changing

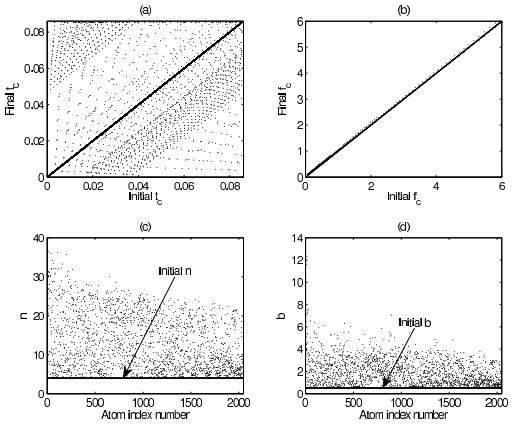

Fig. 8. Parameters of the Gammatone dictionary: the scatter-plots of the parameters t_c and f_c are shown in the top windows (a) and (b) respectively. The initial and final values of n and b are shown in the bottom windows (c) and (d) respectively.

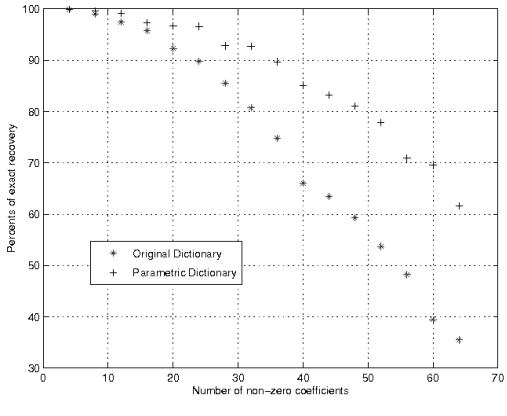

Fig. 9. Exact support recovery of the sparse signals.

the structure of the dictionary, the locations which are not covered by the high energy part of any atom demonstrate its local minimum convergence. It also shows a potential for a more efficient update operator than the gradient descent in Algorithm 2. There exists a shift-invariance structure, with different step size for each frequency band, in the initial parametric dictionary, which disappears in the designed dictionary. If the time-shift is one of the parameters in the parametric dictionary, such a structure can then be preserved. Such a parametric dictionary is not column separable. Designing a structured parametric dictionary, e.g. shift-invariant dictionary, is left for a future work.

The parameter set γ is selected intuitively in this experiment. To show the contribution of each parameter in the dictionary design, we show the initial and the final values of the parameters in Fig.8. The scatter-plots of t_c and f_c are shown in part (a) and (b). Note f_c has not have changed significantly by the dictionary design, so it could be kept fixed to reduce the computational cost. This simulation is also initialized with some fixed values for n and b. The final values of these two parameters are shown in Fig.8.c and Fig.8.d respectively. These plots show significant changes in the values of n and b, which demonstrates the importance of correct selection of n and b in each Gammatone atom.

2) Exact sparse recovery and sparse approximation: In this part we demonstrate the advantages of the parametric dictionary design in terms of exact sparse recovery [25] and sparse approximation. The exact recovery condition (ERC) [25] is studied in a worst-case setting. In this setting when a dictionary satisfies ERC, *any* k-sparse representation can exactly be recovered using (O)MP or BP. In practical applications, an average case analysis is more relevant [52], especially when the probability of the failure is very low. Here, by an experiment, it has been shown that the proposed algorithm improves the average exact recovery. We synthetically generate the sparse coefficient vectors, with different sparsity, and plot the percentages of the exact recovery for those sparse vectors. The location of the non-zero coefficients are selected uniformly at random and the PDF of the magnitudes are selected to be Gaussian with zero mean. The matching pursuit algorithm is used to find the sparse approximation. The rate of exact support recovery is calculated by the ratio of the number of correctly found non-zero coefficient index sets to the number of cases in which at least one location of the zero coefficient is set to non-zero. We run the simulations 1000 times. We have shown this ratio as the percentage of exact recovery in Fig. 9. It is clear that the design method has improved the exact recovery ratio.

For sparse approximation applications, we are more interested to have a dictionary that, if it fails to satisfy exact recovery [25], still gives a sparse approximation for a given class of signals. Therefore as the second experiment, we compare the decay rates of the residual error when the MP is used for sparse approximation [27]. We use an audio signal taken from more than 8 hours recorded from BBC Radio 3, which mostly plays classical music. We first down-sample by a factor of 4 and sum the stereo channels to make a mono signal with 12K samples per second. We use the original Gammatone and the parametric designed dictionaries to approximate 100 randomly selected blocks, each with the length of 1024 samples. The average decay rate of the residual errors, in logarithmic scales, are shown in Fig. 10. This rate directly influences the performance of sparse approximation methods. That is, we can better approximate the signal with fewer coefficients using a high residual error decay rate dictionary. In Fig. 10, although the curves start with the same slope, after a few iterations, here 10, the designed dictionary shows a clear advantage.

V. Conclusion

The sparse approximation methods successfully approximate a class of signals with a set of sparse coefficient vectors,

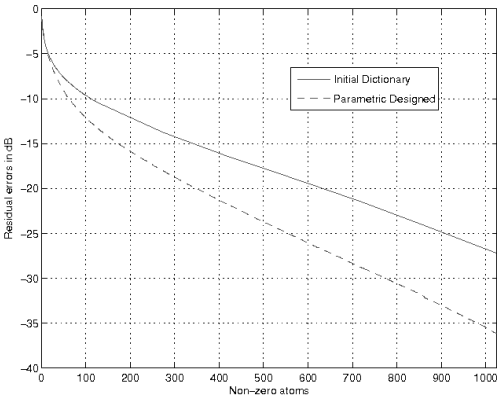

Fig. 10. The residual error using matching pursuit for sparse approximation of the audio signal.

when an appropriate generative model is given. In this paper we have introduced a method to design such a model, which is independent of the signal. A criterion based on an important feature for the success of sparse approximation methods is proposed. A priori knowledge about the signal was included by using parametric functions. In this framework we have shown that the dictionary design problem is to find an optimal set of parameters. This problem can in general not be solved exactly. Fortunately an approximate solution can be found using the proposed method. In some simulations we showed that A) the given method can find an appropriate set of parameters for the given case study and B) the designed dictionary showed promising performance advantages in terms of exact recovery and sparse approximation of audio signals. What we have shown in this paper is a first step in the design of parametric dictionaries. Extra constraints, such as shift-invariance, quasi-incoherence, data dependence, to have tree structures or structures for fast implementation, could be imposed. However, this has been left for future work.

APPENDIX A
CONVERGENCE STUDY OF THE ALGORITHM

To study the convergence of the algorithm, we first show that Algorithm 1, for any parameter update algorithm (line 6), reduces or keeps the same the objective function. The objective is lower bounded by zero and the algorithm prevents the existence of a continuum of fixed points, which guarantee the stability of Algorithm 1. In the next step we show that when $\mathbf{D}_\Gamma$ is a differentiable function on a compact Υ, the sequence generated by the algorithm becomes as close as possible to a set of fixed points.

A. Definition of a surrogate optimization problem

The objective function in (3) depends on two variables, which makes the convergence analysis more difficult, if we want to use the continuity of the objective in the analysis. Here we define a surrogate objective for (3), which has a single variable, to show the convergence of Algorithm 1 to a set of fixed points. Let $\Gamma^* \in \Upsilon$ and $\mathbf{G}^* \in \Lambda^N$ be a solution pair of (3) and $\mathbf{G}^*_\Gamma = \mathbf{D}^T_{\Gamma^*}\mathbf{D}_{\Gamma^*}$. Then $\mathbf{G}^* = \mathcal{P}_{\Lambda^N}\mathbf{G}^*_\Gamma$, which suggests the optimization problem (3) can be replaced by the following problem based on Γ, as the only parameter,

$$\begin{aligned}&\min_{\Gamma\in\Upsilon}\ \Phi_S(\Gamma),\\ &\Phi_S(\Gamma) = \|\mathbf{G}_\Gamma - \mathcal{P}_{\Lambda^N}\mathbf{G}_\Gamma\|_F\\ &\qquad = \Big(\sum_{i\neq j}(|\{g_\Gamma\}_{i,j}| - \mu_G)^2 + \sum_{i=j}(\{g_\Gamma\}_{i,j} - 1)^2\Big)^{1/2},\end{aligned} \tag{11}$$

where $|\{g_\Gamma\}_{i,j}|$ is the absolute value of the (i,j) element of $\mathbf{G}_\Gamma = \mathbf{D}^T_\Gamma\mathbf{D}_\Gamma$. The problems (3) and (11) share common solutions. Therefore one can optimize (11) to find the solution(s) of (3). Although the surrogate objective is a continuous function of Γ ($\Phi_S \in \text{class}\, C^0$), a difficulty with the optimization of the surrogate objective directly is that it is non-differentiable. We only use the surrogate optimization problem to show the convergence of the proposed algorithm.

B. Convergence analysis of Algorithm 1 using the surrogate optimization problem

We now show that the proposed algorithm reduces the surrogate objective at each parameter update, using the following proposition.

Proposition 1: Let $\mathbf{G}_{\Gamma_k} = \mathbf{D}^T_{\Gamma_k}\mathbf{D}_{\Gamma_k}$ be the Gram matrix of the dictionary at k^{th} iteration. The Algorithm 1 reduces, or keeps the same, $\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F$ in each update of the parameters ($\Gamma_k \to \Gamma_{k+1}$), where $\mathcal{P}_{\Lambda^N}$ is the operator of orthogonal projection onto Λ^N.

Proof: Let $\mathbf{G}_{P_{k+1}}$ be an abbreviation for $\mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}$, which is found by using (4). Using the parameter update step (line 6) and the fact that $\mathbf{G}_{R_{k+1}} = \alpha_k\mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$,

$$\begin{aligned}&\alpha_k\|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F\\ &\quad = \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\\ &\quad \geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{R_{k+1}}\|_F\\ &\quad = \|\mathbf{G}_{\Gamma_{k+1}} - \alpha_k\mathbf{G}_{P_{k+1}} - (1-\alpha_k)\mathbf{G}_{\Gamma_k}\|_F\\ &\quad = \|(\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}) - (1-\alpha_k)(\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}})\|_F\\ &\quad \geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}\|_F - (1-\alpha_k)\|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F\ ,\end{aligned}$$

where we used the triangular inequality to derive the last inequality. This provides us the following inequalities,

$$\begin{aligned}\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F &\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F\\ &\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_{k+1}}\|_F,\end{aligned} \tag{12}$$

where the last inequality is easily derived by using the definition of the projection in Hilbert space. ■

Prop. 1, with the facts that the objective is lower bounded by zero and there exists no continuum of fixed points, guarantees stability of Algorithm 1, due to Lyapunov's second theorem.

Let $\text{class}\, C^1$ consist of all continuously differentiable functions. The following two Lemmata are needed to show the convergence of Algorithm 1 to a set of fixed points.

Lemma 1: Let $\mathbf{D}_\Gamma : \Upsilon \rightarrow \mathbb{R}^{d\times N} \in \text{class}\, C^1$ and Υ be compact. The epigraph of the objective (11) at an admissible Γ_0 is compact.

Proof: When the parametric dictionary $\mathbf{D}_\Gamma$ is differentiable on Υ, the objective function in (11) is continuous. The continuity of the surrogate objective function and the compactness of Υ prove the compactness of $\text{epi}\,\Phi_S$ at an admissible point Γ_0 [40]. ■

Due to the Bolzano-Weierstrass theorem, Algorithm 1 has a non-empty set of accumulation points. We now reformulate Lemma 1 in [46] for a more general (including asymptotically non-regular[7]) sequence. Although the proof is the same, the set of accumulation points can be dis-connected, when the sequence is not asymptotically regular.

Lemma 2: Let $\{\Gamma_n\}_{n\in\mathbb{N}}$ be an infinite sequence in a compact set Σ and T be the set of its accumulation points then, $\forall\epsilon > 0, \exists N \in \mathbb{N}$ such that for all $n > N, \exists\Gamma^\ddagger \in T, \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$

Proof: Let S be an ϵ-neighborhood of T and S_c be its complement in Σ. Σ is compact, thus S_c is also compact. Because S is a neighborhood of T, there is no accumulation point Γ in S_c. If $\{\Gamma_n\}$ has infinite many points in S_c, then it has a converging subsequence and at least one accumulation point in S_c. This contradicts the fact that there is no accumulation point in S_c. Therefore $\exists N : \Gamma_n \in S,\ \forall n > N$. On the other hand the fact that S being an ϵ-neighborhood implies that for all $n > N$, $\exists\Gamma^\ddagger \in T : \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$. ■

Theorem 3: Let $\mathbf{D}_\Gamma : \Upsilon \rightarrow \mathbb{R}^{d\times N} \in \text{class}\, C^1$. The Algorithm 1 converges to a set of fixed points by starting from $\Gamma_0 \in \Upsilon$, where Υ is a compact set.

Proof: Due to Lemma 1 the epigraph of the surrogate objective at Γ_0 ($\text{epi}\,\Phi_S(\Gamma_0)$) is compact. The Proposition 1 shows that the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ is in $\text{epi}\,\Phi_S(\Gamma_0)$. The convergence of the algorithm to a non-empty set of accumulation points is guaranteed using Lemma 2. Line 6 of Algorithm 1 prevents the existence of a continuum of accumulation points. Therefore the accumulation points are fixed points. ■

APPENDIX B
GRADIENT OF THE GAMMATONE DICTIONARY

We calculate the gradient of the parametric Gammatone dictionary with the generative function (8) in this appendix. Let $\mathbf{D}_\Gamma \in \mathbb{R}^{d\times N}$ and $\Gamma \in \mathbb{R}^{4\times N}$. The i^{th} column of $\mathbf{D}_\Gamma$ is a function of the i^{th} column of Γ, $\mathbf{d}_{\gamma_i}$. The rank of $\nabla_\Gamma \mathbf{D}_\Gamma$ is 4 and we represent it by a tensor in $\mathbb{R}^{4\times d\times N}$. Each submatrix of this tensor (fixing the third index) is the gradient of the corresponding atom in $\mathbf{D}_\Gamma$. Therefore we only need to calculate the gradient of $\mathbf{d}_{\gamma_i}$ based on γ_i. Because $\mathbf{d}_\gamma$ is calculated using (10), we only need to derive a formulaton for the gradients of $g_\gamma(t)$ based on t_c, f_c, n and b, followed by sampling t.

$$\frac{\partial g_\gamma}{\partial t_c} = -a((n-1)t_s^{n-2}\cos 2\pi f_c t_s + 2\pi b B t_s^{n-1}\cos 2\pi f_c t_s + 2\pi f_c t_s^{n-1}\sin(2\pi f_c t_s))e^{-2\pi b B t_s}$$

[7]A sequence $\{a_k\}_{k\in\mathbb{N}}$ in a normed space is called asymptotically regular when $\lim_{k\rightarrow\infty}\|a_k - a_{k-1}\| = 0$

$$\frac{\partial g_\gamma}{\partial f_c} = a t_s^{n-1}(-2\pi t_s \frac{dB}{df_c}\cos(2\pi f_c t_s) - 2\pi t_s \sin(2\pi f_c t_s))e^{-2\pi b B t_s}$$

$$\frac{\partial g_\gamma}{\partial n} = a\ln(t_s)\ t_s^{n-1}e^{-2\pi b B t_s}\cos(2\pi f_c t_s)$$

$$\frac{\partial g_\gamma}{\partial b} = -2\pi a B t_s^n e^{2\pi b B t_s}\cos(2\pi f_c t_s)$$

where $t_s = t - t_c$ and $\frac{dB}{df_c} = 1/Q$. Some researchers have proposed more complex formulations for B. In this case, one can substitute B and $\frac{dB}{df_c}$ in the above formulas to find the gradient.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their constructive comments and L. Jacques for bringing their attention to [30] and [31]. MY acknowledges the hospitality and support of E. Ravelli at the Institut Jean le Rond d'Alembert-LAM, during his visit, at the start of this research. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

REFERENCES

[1] R. Patterson, M. Allerhand, and C. Giguere, "Time-domain modeling of peripheral auditory processing: a modular architecture and a software platform," *Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1890–1894, 1995.

[2] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.

[3] J. Tropp, "Topics in sparse approximation," Ph.D. dissertation, University of Texas at Austin, 2004.

[4] G. Davis, "Adaptive nonlinear approximations," Ph.D. dissertation, New York University, 1994.

[5] S. Mallat and Z. Zhang, "Matching pursuits with time frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[6] T. Blumensath and M. Davies, "Gradient pursuits," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2370–2382, 2008.

[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[8] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[9] B. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.

[10] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, 2003.

[11] S. Sardy, A. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.

[12] E. Candes and L. Demanet, "The curvelet representation of wave propagators is optimally sparse," *Communications on Pure and Applied Mathematics*, vol. 58, no. 11, pp. 1472–1528, 2005.

[13] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[14] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.

[15] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, and T. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.

[16] M. Aharon, E. Elad, and A. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[18] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[19] A. Katsiamis, E. Drakakis, and R. Lyon, "Practical Gammatone-like filters for auditory processing," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, 2007, Article ID 63685.

[20] T. Irino and R. Patterson, "A time domain, level dependent auditory filter: the Gammachirp," *Journal of the Acoustical Society of America*, vol. 101, pp. 412–419, 1997.

[21] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the Gammatone function," APU Report, Tech. Rep., 1988.

[22] M. Turner, G. Gerstein, and R. Bajcsy, "Underestimation of visual texture slant by human observers: a model," *Biological Cybernetics*, vol. 65, no. 4, pp. 215–226, 2004.

[23] S. Strahl and A. Mertins, "Sparse gammatone signal model optimized for English speech does not match the human auditory filters," *Brain Research*, vol. 1220, pp. 224–233, 2008.

[24] D. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[25] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[26] ——, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[27] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. on Information Theory*, vol. 52, no. 1, pp. 255–261, 2006.

[28] J. Tropp, I. Dhillon, R. Heath Jr., and T. Strohmer, "Designing structural tight frames via an alternating projection method," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.

[29] M. Sustik, J. Tropp, I. Dhillon, and R. Heath, "On the existence of equiangular tight frames," *Linear Algebra and Its Applications*, vol. 426, no. 2-3, pp. 619–635, 2007.

[30] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Trans. on Signal Processing*, vol. 49, no. 5, pp. 994–1001, 2001.

[31] L. Jacques and C. De Vleeschouwer, "A geometrical study of matching pursuit parametrization," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 2835–2848, 2008.

[32] T. Strohmer and R. Heath, "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[33] A. Lyapunov, *Stability of motion*. Academic Press, 1966.

[34] H. Stark and Y. Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*. John Wiley & Sons, Inc, 1998.

[35] H. Hein and S. Hundal, "An alternating projection that does not converge in norm," *Nonlinear Analysis*, vol. 57, no. 1, pp. 35–61, 2004.

[36] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplementary no. 1*, pp. 205–237, 1984.

[37] A. Gunawardana and W. Byrne, "Convergence theorems for generalized alternating minimization procedures," *The Journal of Machine Learning Research*, vol. 6, pp. 2049 – 2073, 2005.

[38] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2005 (v2007.09.17), Palo Alto, CA.

[39] R. Fletcher, *Practical Methods of Optimization*. John Wiley and Sons: Chichester and New York, 1987.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[41] T. Apostol, *Mathematical Analysis: A Modern Approach to Advanced Calculus*. Addison-Wesley, 1974.

[42] M. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.

[43] E. Ravelli, G. Richard, and L. Daudet, "Union of MDCT bases for audio coding," 2008, accepted for publication in IEEE Trans. on Audio, Speech and Language Processing.

[44] E. Smith and M. S. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Computation*, vol. 17, no. 1, pp. 19–45, 2005.

[45] M. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, 2002.

[46] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[47] R. Pichevar, H. Najaf-Zadeh, and L. Thibault, "A biologically-inspired low-bit-rate universal audio coder," in *Audio Engineering Society Convention, Vienna, Austria*, 2007.

[48] B. R. Glasberg and B. C. J. Moore, "Derivative of auditory filter shapes from notched-noise data," *Hearing Research*, vol. 47, pp. 103–108, 1990.

[49] M. Slaney, "Lyon's cochlear model," Apple Computer, Tech. Rep., 1988.

[50] ——, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Tech. Rep., 1993.

[51] S. Abellah and M. Plumbley, "If the independent components of natural images are edges, what are the independent components of natural sounds?" in *Proc. of the International Conference on Independent Component Analysis and Blind Source Separation (ICA)*, 2001.

[52] R. Gribonval and K. Schnass, "Some recovery conditions for basis learning by L1-minimization," in *International Symposium on Communications, Control and Signal Processing, ISCCSP*, 2008.

Mehrdad Yaghoobi (S'98-M'09) received the BSc. and MSc. in electrical and biomedical engineering in 1999 and 2002 from the University of Tehran and Sharif University of Technology, respectively. He started his PhD at Queen Mary University of London in December 2005, before he moved to the University of Edinburgh to accompany his supervisor in April 2006. He is now persuing the PhD degree in the Institute for Digital Communications (IDCom) at the University of Edinburgh. His current research interests include sparse approximation, dictionary selection, compressed sensing and audio modelling/coding.

Laurent Daudet (M'04) received the M.S. degree in statistical and nonlinear physics from the cole Normale Superieure, Paris, France, in 1997 and the Ph.D. degree in mathematical modeling from the Universite de Provence, Marseille, France, in audio signal representations, in 2000. In 2001 and 2002, he was a Marie Curie Post-doctoral Fellow with the Centre for Digital Music at Queen Mary, University of London, London, U.K. Since 2002, he has been working as an Assistant Professor at the UPMC Univ. Paris 06, Paris, France, where he joined the Laboratoire d'Acoustique Musicale, now part of Institut Jean Le Rond d'Alembert. His research interests include audio coding, time-frequency and time-scale transforms, and sparse representations for audio.

Mike E. Davies (M'00) received the B.A. (Hons.) degree in engineering from Cambridge University, Cambridge, U.K., in 1989 and the Ph.D. degree in nonlinear dynamics and signal processing from University College London, London (UCL), U.K., in 1993. Mike Davies was awarded a Royal Society Research Fellowship in 1993 and was an Associate Editor for IEEE Transactions in Speech, Language and Audio Processing, 2003-2007.

He currently holds the Jeffrey Collins SHEFC funded chair in Signal and Image Processing at the University of Edinburgh. His current research interests include: sparse approximation, compressed sensing and their applications.

References

[ABB+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *Lapack user's guide, third edition*, Society for Industrial and Applied Mathematics (SIAM), 1999.

[AD05] T. Adeyemi and M.E. Davies, *Sparse representations of images using overcomplete complex wavelets*, IEEE Workshop on Statistical Signal Processing (SSP), July 2005, pp. 805–809.

[AE08] M. Aharon and M. Elad, *Sparse and redundant modeling of image content using an image-signature-dictionary*, SIAM Journal on Imaging Sciences **1** (2008), no. 3, 228–247.

[AEB06] M. Aharon, E. Elad, and A.M. Bruckstein, *K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation*, IEEE Trans. on Signal Processing **54** (2006), no. 11, 4311–4322.

[AMN+99] O.K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, *Video compression using matching pursuits*, IEEE Transactions on Circuits and Systems for Video Technology **9** (1999), no. 1, 123–143.

[AP01] S.A. Abellah and M.D. Plumbley, *If the independent components of natural images are edges, what are the independent components of natural sounds?*, Proc. of the International Conference on Independent Component Analysis and Blind Source Separation (ICA), 2001.

[Apo74] T.M. Apostol, *Mathematical analysis: A modern approach to advanced calculus*, Addison-Wesley, 1974.

[BBC09] S. Becker, J. Bobin, and E. J. Candes, *Nesta: a fast and accurate first-order method for sparse recovery*, submitted for publication, 2009.

[BD97] R. L. Burden and J. Douglas Faires, *Numerical analysis*, Brooks/Cole, 1997.

[BD06] T. Blumensath and M.E. Davies, *Sparse and shift-invariant representations of music*, IEEE Trans. on Audio, Speech and Language Processing **14** (2006), no. 1, 50–57.

[BD07] ———, *Monte Carlo methods for adaptive sparse approximations of time-series*, IEEE Transactions on Signal Processing **55** (2007), no. 9, 4474–4486.

[BD08a] ———, *Gradient pursuits*, IEEE Trans. on Signal Processing **56** (2008), no. 6, 2370–2382.

[BD08b] ———, *Iterative thresholding for sparse approximations*, Journal of Fourier Analysis and Applications **14** (2008), no. 5, 629–654.

[BD09] ———, *Iterative hard thresholding for compressed sensing*, Accepted for publication in Applied and Computational Harmonic Analysis, 2009.

[BF07] J.M. Bioucas-Dias and M.A.T. Figueiredo, *A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration*, IEEE Transactions on Image Processing **16** (2007), no. 12, 2992–3004.

[BL08] K. Bredies and D. A. Lorenz, *Linear convergence of iterative soft-thresholding*, Journal of Fourier Analysis and Applications **14** (2008), no. 5, 813 – 837.

[Blu06] T. Blumensath, *Bayesian modelling of music: Algorithmic advances and experimental studies of shift-invariant sparse coding*, Ph.D. thesis, Queen Mary, University of London, 2006.

[Bru77] P. Brucker, *On the complexity of clustering problems*, Workshop on Optimization and Operations Research, 1977, pp. 45–54.

[BST98] A. Bruce, S. Sardy, and P. Tseng, *Block coordinate relaxation methods for nonparametric signal denoising*, Proceedings of the SPIE - The International Society for Optical Engineering, 1998, pp. 75–86.

[BT08] A. Beck and M. Teboulle, *A fast iterative "shrinkage-thresholding" algorithm for linear inverse problems*, Tech. report, Technion - Israel Institute of Technology, 2008.

[BV04] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[BYD07] T. Blumensath, M. Yaghoobi, and M.E. Davies, *Iterative hard thresholding and l0 regularisation*, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, vol. 3, April 2007, pp. 877–880.

[CBL89] S. Chen, S.A. Billings, and W. Luo, *Orthogonal least squares methods and their application to non-linear system identification*, International Journal of Control **50** (1989), no. 5, 18731896.

[CD05] E.J. Candes and L. Demanet, *The curvelet representation of wave propagators is optimally sparse*, Communications on Pure and Applied Mathematics **58** (2005), no. 11, 1472–1528.

[CDS98] S.S. Chen, D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing **20** (1998), no. 1, 33–61.

[CH06] J. Chen and X. Huo, *Theoretical results on sparse representations of multiple measurement vectors*, IEEE Trans. on Signal Processing **54** (2006), no. 12, 4634–4643.

[Cha07] R. Chartrand, *Exact reconstruction of sparse signals via nonconvex minimization*, Signal Processing Letters, IEEE **14** (2007), no. 10, 707–710.

[Com94] P. Comon, *Independent component analysis, a new concept?*, Signal Processing **36** (1994), 287–314.

[CREKD05] S.F. Cotter, B.D. Rao, K. Engan, and K. Kreutz Delgado, *Sparse solutions to linear inverse problems with multiple measurement vectors*, IEEE Trans. on Signal Processing **53** (2005), no. 7, 2477–2488.

[CRT06a] E. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. on Information Theory **52** (2006), no. 2, 489–509.

[CRT06b] _____, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics **59** (2006), no. 8, 1207–1223.

[CT84] I. Csiszar and G. Tusnady, *Information geometry and alternating minimization procedures*, Statistics and Decisions, Supplementary no. 1 (1984), 205–237.

[CT90] P. L. Combettes and H. J. Trussell, *Method of successive projections for finding a common point of sets in metric spaces*, Journal of Optimization Theory and Applications **67** (1990), no. 3, 487–507.

[CT91] T.M. Cover and J.A. Thomas, *Elements of information theory*, Wiley Interscience, 1991.

[Cve03] Z. Cvetkovic, *Resilience properties of redundant expansions under additive noise and quantization*, IEEE Trans. on Information Theory **49** (2003), no. 3, 644656.

[CW05] P.L. Combettes and V.R. Wajs, *Signal recovery by proximal forward-backward splitting*, SIAM Journal on Multiscale Modeling and Simulation **4** (2005), 1168–1200.

[CWB08] E. J. Candes, M. B. Wakin, and S. P. Boyd, *Enhancing sparsity by reweighted l1 minimization*, Journal of Fourier Analysis and Applications **14** (2008), no. 5, 877–905.

[Dat09] J. Dattorro, *Convex optimization and Euclidean distance geometry*, Meboo Publishing, 2009, (v2009.06.18), Palo Alto, CA.

[Dau80] J.G. Daugman, *Two-dimensional spectral analysis of cortical receptive field profile*, Vision Research **20** (1980), 847–856.

[Dau92] I. Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, 1992.

[DD06] M.E. Davies and L. Daudet, *Sparse audio representations using the MCLT*, Signal Processing **86** (2006), no. 3, 457–470.

[DDD04] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math **57** (2004), 1413–1541.

[DDFC08] I. Daubechies, R De Vore, M. Fornasier, and Sinan Gunturk C., *Iteratively reweighted least squares minimization for sparse recovery*, to appear in Comm. Pure Appl. Math., 2008.

[DE03] D. Donoho and M. Elad, *Optimally-sparse representation in general (non-orthogonal) dictionaries via l1 minimization*, Proceedings of the National Academy of Sciences **100** (2003), 2197–2202.

[DeV98] R.A. DeVore, *Nonlinear approximation*, Acta Numerica **7** (1998), 51150, Cambridge Univ. Press, Cambridge.

[DFK+04] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, *Clustering large graphs via the singular value decomposition*, Machine Learning **56** (2004), no. 1, 9–33.

[DFL08] I. Daubechies, M. Fornasier, and I. Loris, *Accelerated projected gradient method for linear inverse problems with sparsity constraints*, Journal of Fourier Analysis and Applications **14** (2008), no. 5, 764 – 792.

[DG09] M.E. Davies and R. Gribonval, *Restricted isometry constants where ℓ^p sparse recovery can fail for $0 < p \leq 1$*, IEEE Transactions on Information Theory **55** (2009), no. 5, 2203–2214.

[DGV06] O. Divorra Escoda, L. Granai, and P. Vandergheynst, *On the use of a priori information for sparse signal approximations*, IEEE Trans. on Signal Processing **54** (2006), no. 9, 3468–3482.

[DH01] D.L. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. on Information Theory **47** (2001), no. 7, 2845–2862.

[DJ94] D.L. Donoho and J.M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika **81** (1994), no. 3, 425–455.

[DMA97] G. Davis, S. Mallat, and M. Avellaneda, *Adaptive greedy approximations*, Constructive Approximation **13** (1997), no. 1, 57–98.

[Don95] D.L. Donoho, *Denoising by soft-thresholding*, IEEE Trans. on Information Theory **41** (1995), no. 3, 613–627.

[Don04a] D. Donoho, *For most large underdetermined systems of linear equations, the minimal ℓ_1 -norm solution is also the sparsest solution*, Communications on Pure and Applied Mathematics **59** (2004), no. 6, 797–829.

[Don04b] D.L. Donoho, *Neighborly polytopes and sparse solutions of underdetermined linear equations.*, Tech. report, Statistics Department, Stanford University, 2004.

[Don06] D. Donoho, *Compressed sensing*, IEEE Trans. on Information Theory **52** (2006), no. 4, 1289–1306.

[DT02] L. Daudet and B. Torresani, *Hybrid representations for audiophonic signal encoding*, Signal Processing, special issue on Coding Beyond Standards **82** (2002), no. 11, 1595–1617.

[DTDS06] D. Donoho, Y Tsaig, I. Drori, and J. Starck, *Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit*, Tech. report, Stanford University, 2006.

[DZ03] C. De Vleeschouwer and A. Zakhor, *In-loop atom modulus quantization for matching pursuit and its application to video coding*, IEEE Trans. on Image Processing **12** (2003), no. 10, 1226– 1242.

[EA06] M. Elad and M. Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Trans. on Image Processing **15** (2006), no. 12, 3736–3745.

[EAH99a] K. Engan, S.O. Aase, and J. Hakon-Husoy, *Method of optimal directions for frame design*, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1999.

[EAH99b] K. Engan, S.O. Aase, and J.H. Husoy, *Frame based signal compression using method of optimal directions (MOD)*, IEEE International Symposium on Circuits and Systems, vol. 4, Jul 1999, pp. 1–4 vol.4.

[EB02] M. Elad and A.M. Bruckstein, *A generalized uncertainty principle and sparse representation in pairs of bases*, IEEE Trans. on Information Theory **48** (2002), no. 9, 2558–2567.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, *Least angle regression*, Annual of Statistics **32** (2004), no. 2, 407–499.

[Ela06] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Trans. on Information Theory (2006), no. 12, 5559–5569.

[EMR07] M. Elad, P. Milanfar, and R. Rubinstein, *Analysis versus synthesis in signal priors*, Inverse Problems **23** (2007), no. 3, 947–968.

[EMSZ07] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, *A wide-angle view at iterated shrinkage algorithms*, Proc. SPIE, Vol. 6701, 2007.

[EMZ07] M. Elad, B. Matalon, and M. Zibulevsky, *Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization*, Applied and Computational Harmonic Analysis **23** (2007), no. 3, 346–367.

[Eng00] K. Engan, *Frame based signal representation and compression*, Ph.D. thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, 2000.

[ERK99] K. Engan, B.D. Rao, and K Kreutz-Delgado, *Frame design using FOCUSS with method of optimal directions (mod)*, Norwegian Signal Processing Symposium, (Asker, Norway, 9-11 Sept. 1999.) Tronheim, Norway: NORSIG, 1999, pp. 65–9.

[ESH07] K. Engan, K. Skretting, and J. H. Husoy, *Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation*, Digital Signal Processing **17** (2007), no. 1, 32 – 49.

[ESQD05] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (mca)*, Journal on Applied and Computational Harmonic Analysis **19** (2005), 340–358.

[FBSJ08] H. Firouzi, M. Babaie-Zadeh, A.G. Sahebi, and C. Jutten, *A first step to convolutive sparse representation*, IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), 31 2008-April 4 2008, pp. 1921–1924.

[FH79] J.C. Fiorot and P. Huard, *Composition and union of general algorithms of optimization*, Mathematical Programming Study **10** (1979), 69–85.

[FL09] S. Foucart and M.J. Lai, *Sparsest solutions of underdetermined linear systems via ℓq-minimization for $0 < q \leq$*, Applied and Computational Harmonic Analysis **26** (2009), no. 3, 395–407.

[Fle87] R. Fletcher, *Practical methods of optimization*, John Wiley and Sons: Chichester and New York, 1987.

[FN03] M.A.T. Figueiredo and R.D. Nowak, *An EM algorithm for wavelet-based image restoration*, IEEE Trans. on Image Processing **12** (2003), no. 8, 906–916.

[FN05] ———, *A bound optimization approach to wavelet-based image deconvolution*, International Conference on Image Processing (ICIP), 2005, pp. 782–785.

[FNW07] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing **1** (2007), no. 4, 586–597.

[FR08a] M. Fornasier and H. Rauhut, *Iterative thresholding algorithms*, Applied and Computational Harmonic Analysis **25** (2008), no. 2, 187 – 208.

[FR08b] M. Fornasier and H. Rauhut, *Recovery algorithms for vector valued data with joint sparsity constraints*, SIAM Journal of Numerical Analysis **46** (2008), no. 2, 577–613.

[FRGR06] A.K. Fletcher, S. Rangan, V.K. Goyal, and K. Ramchandran, *Denoising by sparse approximation: Error bounds based on rate-distortion theory*, Journal on Applied Signal Processing **10** (2006), 1–19.

[FV01] P. Frossard and P. Vandergheynst, *A posteriori quantized matching pursuit*, IEEE Data Compression Conference, 2001.

[FVFK04] P. Frossard, P. Vandergheynst, R.M. Figueras i Ventura, and M. Kunt, *A posteriori quantization of progressive matching pursuit streams*, IEEE Trans. on Signal Processing **52** (2004), no. 2, 525–535.

[Gab46] D. Gabor, *Theory of communication*, Journal of IEE **93** (1946), 429–457.

[GB03] R. Gribonval and E. Bacry, *Harmonic decomposition of audio signals with matching pursuit*, IEEE Trans. **51** (2003), no. 1, 101– 111.

[GB05] A. Gunawardana and W. Byrne, *Convergence theorems for generalized alternating minimization procedures*, The Journal of Machine Learning Research **6** (2005), 2049 – 2073.

[GG91] A. Gersho and R.M. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, 1991.

[GKK01] V.K. Goyal, J. Kovacevic, and J. A. Kelner, *Quantized frame expansions with erasures*, Applied and Computational Harmonic Analysis **10** (2001), no. 3, 203–233.

[GM90] B. R. Glasberg and B. C. J. Moore, *Derivative of auditory filter shapes from notched-noise data*, Hearing Research **47** (1990), 103–108.

[GMS03] A.C. Gilbert, S. Muthukrishnan, and M.J. Strauss, *Approximation of functions over redundant dictionaries using coherence*, Proc. 14th Annu. ACM-SIAM Symp. on Discrete algorithms, Baltimore, MD, 2003.

[GN03] R. Gribonval and M. Nielsen, *Sparse representations in unions of bases*, IEEE Trans. on Information Theory **49** (2003), no. 12, 3320 – 3325.

[GN07] ———, *Highly sparse representations from dictionaries are unique and independent of the sparseness measure*, Applied and Computational Harmonic Analysis **22** (2007), no. 3, 335–355.

[GPR67] L. G. Gubin, B. T. Polyak, and E. V. Raik, *The method of projections for finding the common point of convex sets*, USSR Computational Mathematics and Mathematical Physics **7** (1967), 124.

[GR97] I.F. Gorodnitsky and B.D. Rao, *Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm*, IEEE Trans. on Signal Processing **45** (1997), no. 3, 600–616.

[Gra06] R.M. Gray, *Toeplitz and circulant matrices: A review*, Now Publishers Inc, 2006.

[GRKN07] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, *Shift-invariant sparse coding for audio classification*, the Twenty-third Conference on Uncertainty in Artificial Intelligence, 2007.

[GS08] R. Gribonval and K. Schnass, *Some recovery conditions for basis learning by L1-minimization*, International Symposium on Communications, Control and Signal Processing, ISCCSP, 2008.

[GS09] R. Gribonval and K. Schnass, *Dictionary identification: Sparse matrix-factorisation via ℓ_1 minimisation*, Available at `http://arxiv.org/pdf/0904.4774`, 2009.

[GTC05] P. Georgiev, F. Theis, and A. Cichocki, *Sparse component analysis and blind source separation of underdetermined mixtures*, Neural Networks, IEEE Transactions on **16** (2005), no. 4, 992–996.

[GV97] V.K. Goyal and M. Vetterli, *Dependent coding in quantized matching pursuit*, Proceedings of the SPIE-Visual Communication and Image Processing, 1997.

[GV06] R. Gribonval and P. Vandergheynst, *On the exponential convergence of matching pursuits in quasi-incoherent dictionaries*, IEEE Trans. on Information Theory **52** (2006), no. 1, 255–261.

[GVL96] G.H. Golub and C.F. Van Loan, *Matrix computations*, Johns Hopkins University Press Baltimore, 1996.

[GVT98] V.K. Goyal, M. Vetterli, and N.T. Thao, *Quantized overcomplete expansions in R^N : Analysis, synthesis and algorithms*, IEEE Trans. on Information Theory **44** (1998), no. 1, 16–31.

[HGT06] K.K. Herrity, A.C. Gilbert, and J.A. Tropp, *Sparse approximation via iterative thresholding*, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2006.

[HH04] H.S. Hein and S. Hundal, *An alternating projection that does not converge in norm*, Nonlinear Analysis **57** (2004), no. 1, 35–61.

[HH07] L. Horesh and E. Haber, *Overcomplete dictionary design by empirical risk minimization*, Submitted, 2007.

[HJ85] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge University Press, 1985.

[HO00] A. Hyvarinen and E. Oja, *Independent component analysis: algorithms and applications*, Neural Networks **13** (2000), no. 4-5, 411 – 430.

[H74] J. A. Hgbom, *Aperture synthesis with a non-regular distribution of interferometric baselines*, Astronomy and Astrophysics Supplement **15** (1974), 417–426.

[IP97] T. Irino and R.D. Patterson, *A time domain, level dependent auditory filter: the Gammachirp*, Journal of the Acoustical Society of America **101** (1997), 412–419.

[JH91] C. Jutten and J. Herault, *Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture*, Signal Processing **24** (1991), no. 1, 1 – 10.

[JLVG06] P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval, *MOTIF: An efficient algorithm for learning translation invariant dictionaries*, ICASSP, 2006, pp. 857–860.

[Jol02] I. T. Jolliffe, *Principal component analysis*, Springer, 2002.

[KDL07] A.G. Katsiamis, E.M. Drakakis, and R.F. Lyon, *Practical Gammatone-like filters for auditory processing*, EURASIP Journal on Audio, Speech, and Music Processing **2007** (2007), Article ID 63685.

[KG06] S. Krstulovic and R. Gribonval, *Mptk: Matching pursuit made tractable*, IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), vol. 3, May 2006, pp. 496–499.

[KKL+07] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale l1-regularized least squares*, IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing **1** (2007), no. 4, 606–617.

[KMR+03] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, *Dictionary learning algorithms for sparse representation*, Neural Comp. **15** (2003), 349–396.

[KR03] N.G. Kingsbury and T. H. Reeves, *Iterative image coding with overcomplete complex wavelet transforms*, Conference on Visual Communications and Image Processing, 2003.

[KRE+99] K. Kreutz-Delgado, B. D. Rao, K. Engan, T.W. Lee, and T. J. Sejnowski, *Convex/schur-convex (CSC) log-priors and sparse coding*, Joint Symposium on Neural Computation, 1999.

[KY03] H. J. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*, 2 ed., Springer, 2003.

[Lan51] L. Landweber, *An iterative formula for Fredholm integral equations of the first kind*, Americam Journal of Mathematics **73** (1951), 615–624.

[Lan04] K Lange, *Optimization*, Springer-Verlag, 2004.

[LBG80] Y. Linde, A. Buzo, and R. Gray, *An algorithm for vector quantizer design*, IEEE Trans. on Communications **28** (1980), no. 1, 84–95.

[LBRN07] H. Lee, A. Battle, R. Raina, and A. Y. Ng, *Efficient sparse coding algorithms*, Advances in Neural Information Processing Systems 19 (B. Schölkopf, J. Platt, and T. Hoffman, eds.), MIT Press, Cambridge, MA, 2007, pp. 801–808.

[Lee94] J. Leeuw, *Block-relaxation algorithms in statistics*, in Information Systems and Data Analysis, ed. H.H. Bock, W. Lenski and M. M. Richter, Berlin: Springer-Verlag, pp. 308-325, 1994.

[Lee96] T.S. Lee, *Image representation using 2D Gabor wavelets*, IEEE Trans. on Pattern Analysis and Machine Intelligence **18** (1996), no. 10, 959–971.

[Les07] S. Lesage, *Apprentissage de dictionnaires structurs pour la modlisation parcimonieuse des signaux multicanaux*, Ph.D. thesis, Universiy of Rennes I, 2007.

[Lew02] M.S. Lewicki, *Efficient coding of natural sounds*, Nature Neuroscience **5** (2002), no. 4, 356–363.

[LGBB05] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, *Learning unions of orthonormal bases with thresholded singular value decomposition*, ICASSP, 2005, pp. 293–296.

[LHY00] K. Lange, D.R. Hunter, and I. Yang, *Optimization transfer using surrogate objective functions*, Journal of Computational and Graphical Statistics **9** (2000), no. 1, 1–20.

[LS98] M. S. Lewicki and T.J. Sejnowski, *Coding time-varying signals using sparse, shift-invariant representations*, Proceedings of the conference on Advances in Neural Information Processing Systems (NIPS), 1998.

[LS00] M.S. Lewicki and T.J. Sejnowski, *Learning overcomplete representations*, Neural Comp **12** (2000), no. 2, 337–365.

[Lya66] A.M. Lyapunov, *Stability of motion*, Academic Press, 1966.

[Mal92] H.S. Malvar, *Signal processing with lapped transforms*, Artech House, Inc. Norwood, MA, USA, 1992.

[Mal99] S. Mallat, *A wavelet tour of signal processing*, second ed., Academic Press, 1999.

[MBJ08] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, *Complex-valued sparse representation based on smoothed L0 norm*, Proceeding of the IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), 2008.

[MBZJ09] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, *A fast approach for overcomplete sparse decomposition based on smoothed l0 norm*, IEEE Trans. on Signal Processing **57** (2009), no. 1, 289–301.

[MLGB08] B Mailhe, S. Lesage, R. Gribonval, and F. Bimbot, *Shift-invariant dictionary learning for sparse representations: extending K-SVD*, the European Signal Processing Conference (EUSIPCO), 2008.

[Moo20] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bulletin of the American Mathematical Society **26** (1920), 394395.

[MP06] L. Mancera and J. Portilla, *l0-norm-based sparse representation through alternate projections*, Proceeding of International Conference on Image Processing (ICIP), 2006, pp. 2089–2092.

[MSE08] J. Mairal, G. Sapiro, and M. Elad, *Learning multiscale sparse representations for image and video restoration*, SIAM Multiscale Modeling and Simulation **7** (2008), no. 1, 214–241.

[MZ93] S. Mallat and Z. Zhang, *Matching pursuits with time frequency dictionaries*, IEEE Trans. on Signal Processing **41** (1993), no. 12, 3397–3415.

[Nat95] B.K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal of Comput **24** (1995), no. 2, 227234.

[Nes83] Y.E. Nesterov, *A method for solving the convex programming problem with convergence rate* $\mathcal{O}(1/k^2)$, Dokl. Akad. Nauk SSSR **269** (1983), 543–547.

[Nes07] Y. Nesterov, *Gradient methods for minimizing composite objective function*, Tech. Report 2007/76, CORE Discussion Paper, 2007.

[NZ00] R. Neff and A. Zakhor, *Modulus quantization for matching-pursuit video coding*, IEEE Trans. on Circuits and Systems for Video Technology **10** (2000), no. 6, 895–912.

[NZ02] R. Neff and A. Zakhor, *Matching pursuit video coding - part I: Dictionary approximation*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), no. 1, 1326.

[OF96] B.A. Olshausen and D.J. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature **13** (1996), no. 381, 607–609.

[OF97] ———, *Sparse coding with an overcomplete basis set: a strategy employed by V1?*, Vision Research **37** (1997), no. 23, 3311–3325.

[OM00] Bruno A. Olshausen and K. Jarrod Millman, *Learning sparse codes with a mixture-of-gaussians prior*, In Advances in Neural Information Processing Systems, MIT Press, 2000, pp. 841–847.

[PAG95] R.D. Patterson, M.H. Allerhand, and C Giguere, *Time-domain modeling of peripheral auditory processing: a modular architecture and a software platform*, Journal of the Acoustical Society of America **98** (1995), no. 4, 1890–1894.

[Pal91] B.P. Palka, *An introduction to complex function theory*, Springer, 1991.

[PE09] M. Protter and M. Elad, *Image sequence denoising via sparse and redundant representations*, IEEE Trans. on Image Processing **18** (2009), no. 1, 27–36.

[Pen55] R. Penrose, *A generalized inverse for matrices*, Mathematical Proceedings of the Cambridge Philosophical Society **51** (1955), no. 03, 406–413.

[Plu06] M. D. Plumbley, *Recovery of sparse representations by polytope faces pursuit*, Independent Component Analysis and Blind Signal Separation, 2006, pp. 206–313.

[Plu07a] M.D. Plumbley, *Dictionary learning for l1-exact sparse coding*, International Conference on Independent Component Analysis and Signal Separation, ICA, 2007.

[Plu07b] ———, *On polar polytopes and the recovery of sparse representations*, IEEE Trans. on Information Theory **53** (2007), no. 9, 3188–3195.

[PM07] J. Portilla and L. Mancera, *l0-based sparse approximation: two alternative methods and some applications*, Proceeding of SPIE, 2007.

[PNHR88] R.D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, *An efficient auditory filterbank based on the Gammatone function*, Tech. report, APU Report, 1988.

[PNT07] R. Pichevar, H. Najaf-Zadeh, and L. Thibault, *A biologically-inspired low-bit-rate universal audio coder*, Audio Engineering Society Convention, Vienna, Austria, 2007.

[PO06] B.A. Pearlmutter and R.K. Olsson, *Linear program differentiation for single-channel speech separation*, Machine Learning for Signal Processing, September 2006, pp. 421–426.

[Pot89] F.A. Potra, *On Q-order and R-order of convergence*, Journal of Optimization Theory and Applications **63** (1989), no. 3, 415–431.

[PRK93] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, *Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition*, Asilomar Conference on Signals, Systems and Computers, 1993, pp. 40–44.

[PSZ08] L. Potter, P. Schniter, and J. Ziniel, *Sparse reconstruction for radar*, SPIE Algorithms for Synthetic Aperture Radar Imagery XV, 2008.

[RBC98] D.W. Redmill, D.R. Bull, and Czerepiński, *Video coding using a fast non-separable matching pursuits algorithms*, IEEE International Conference on Image Processing (ICIP), 1998, pp. 769–773.

[RK99] B.D. Rao and K. Kreutz-Delgado, *An affine scaling methodology for best basis selection*, IEEE Trans. on Signal Processing **47** (1999), no. 1, 187–200.

[RL02] L. Rebollo-Neira and D. Lowe, *Optimized orthogonal matching pursuit approach*, IEEE Signal Processing Letters **9** (2002), no. 4, 137–140.

[RLS09] I. Ramirez, F. Lecumberry, and G. Sapiro, *Sparse modeling with universal priors and learned incoherent dictionaries*, Tech. report, Institute for Mathematics and Its Applications, University of Minnesota, 2009.

[Roc70] R.T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[RRD08a] E. Ravelli, G. Richard, and L. Daudet, *Matching pursuit in adaptive dictionaries for scalable audio coding*, Proceeding of the European Signal Processing Conference (EUSIPCO 2008), 2008.

[RRD08b] ———, *Union of MDCT bases for audio coding*, IEEE Trans. on Audio, Speech, and Language Processing **16** (2008), no. 8, 1361–1372.

[RSV08] H. Rauhut, K. Schnass, and P. Vandergheynst, *Compressed sensing and redundant dictionaries*, IEEE Trans. Inform. Theory **54** (2008), no. 5, 2210–2219.

[Rud76] W. Rudin, *Principles of mathematical analysis*, McGraw-Hill, 1976.

[RZE08] R. Rubinstein, M. Zibulevsky, and M. Elad, *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit*, Tech. report, Technion, April 2008.

[RZE09] ———, *Double sparsity: Learning sparse dictionaries for sparse signal approximation*, To appear, 2009.

[SBT00] S. Sardy, A.G. Bruce, and P. Tseng, *Block coordinate relaxation methods for nonparametric wavelet denoising*, Journal of Computational and Graphical Statistics **9** (2000), no. 2, 361–379.

[SED04] J.L. Starck, M. Elad, and D.L. Donoho, *Redundant multiscale transforms and their application for morphological component analysis*, Journal of Advances in Imaging and Electron Physics **132** (2004), 287–348.

[SED05] J.-L. Starck, M. Elad, and D.L. Donoho, *Image decomposition via the combination of sparse representations and a variational approach*, IEEE Transactions on Image Processing **14** (2005), no. 10, 1570–1582.

[SH03] T. Strohmer and R.W. Heath, *Grassmannian frames with applications to coding and communication*, Applied and Computational Harmonic Analysis **14** (2003), no. 3, 257–275.

[SL05] E. Smith and M. S. Lewicki, *Efficient coding of time-relative structure using spikes*, Neural Computation **17** (2005), no. 1, 19–45.

[SL06] E.C. Smith and M. S. Lewicki, *Efficient auditory coding*, Nature **439** (2006), 978–982.

[Sla88] M. Slaney, *Lyon's cochlear model*, Tech. report, Apple Computer, 1988.

[Sla93] ———, *An efficient implementation of the Patterson-Holdsworth auditory filter bank*, Tech. report, Apple Computer, 1993.

[SM03] E. Suli and D. Mayers, *An introduction to numerical analysis*, Cambridge University Press, 2003.

[SM08] S. Strahl and A. Mertins, *Sparse Gammatone signal model optimized for English speech does not match the human auditory filters*, Brain Research **1220** (2008), 224–233.

[SO03] P. Sallee and B. A. Olshausen, *Learning sparse multiscale image representations*, Advances Neural Information Processing Systems (NIPS), vol. 15, 2003.

[STDH07] M. Sustik, J.A. Tropp, I.S. Dhillon, and R.W. Heath, *On the existence of equiangular tight frames*, Linear Algebra and Its Applications **426** (2007), no. 2-3, 619–635.

[SV08] K. Schnass and P. Vandergheynst, *Dictionary learning based dimensionality reduction for classification*, ISCCSP'08, 2008.

[SY98] H. Stark and Y. Yang, *Vector space projections: A numerical approach to signal and image processing, neural nets and optics*, John Wiley & Sons, Inc, 1998.

[SY09] R. Saab and O. Yilmaz, *Sparse recovery by non-convex optimization – instance optimality*, to appear in Applied and Computational Harmonic Analysis, 2009.

[TDHJS05] J.A. Tropp, I.S. Dhillon, R.W. Heath Jr., and T. Strohmer, *Designing structural tight frames via an alternating projection method*, IEEE Trans. on Information Theory **51** (2005), no. 1, 188–209.

[Tem03] V. N. Temlyakov, *Nonlinear methods of approximation*, Foundations of Computational Mathematics **3** (2003), 33–107.

[TGB04] M.R. Turner, G.L. Gerstein, and R. Bajcsy, *Underestimation of visual texture slant by human observers: a model*, Biological Cybernetics **65** (2004), no. 4, 215–226.

[Tib96] R. Tibshirani, *Regression shrinkage and selection via the Lasso*, Journal of Royal Statistical Society Series B, **58** (1996), 267288.

[Tro04a] J.A. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. on Information Theory **50** (2004), no. 10, 2231–2242.

[Tro04b] ———, *Topics in sparse approximation*, Ph.D. thesis, University of Texas at Austin, 2004.

[Tro06a] ———, *Algorithms for simultaneous sparse approximation. part II: Convex relaxation*, Signal Processing **86** (2006), no. 3, 589–602.

[Tro06b] ———, *Just relax: Convex programming methods for identifying sparse signals*, IEEE Trans. on Information Theory **52** (2006), no. 3, 1030–1051.

[TW09] J.E. Tropp and S.J. Wright, *Computational methods for sparse solution of linear inverse problems*, Tech. Report 2009-01, California Institute of Technology, March 2009.

[vBF08] E. van den Berg and M.P. Friedlander, *Probing the pareto frontier for basis pursuit solutions*, SIAM Journal on Scientific Computing **31** (2008), no. 2, 890–912.

[VCFW08] K.R. Varshney, M. Cetin, J.W. Fisher, and A.S. Willsky, *Sparse representation in structured dictionaries with application to synthetic aperture radar*, IEEE Trans. on Signal Processing **56** (2008), no. 8, 3548 – 3561.

[VMS05] R. Vidal, Yi M., and S. Sastry, *Generalized principal component analysis (GPCA)*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005), no. 12, 1945–1959.

[VT09] B. Vikrham Gowreesunker and A. H. Tewfik, *A shift tolerant dictionary training method*, Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09), 2009.

[WNF09] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, Signal Processing, IEEE Transactions on **57** (2009), no. 7, 2479–2493.

[WR04] D.P. Wipf and B.D. Rao, *Sparse bayesian learning for basis selection*, IEEE Transactions on Signal Processing **52** (2004), no. 8, 2153–2164.

[YBD07] M. Yaghoobi, T. Blumensath, and M. Davies, *Quantized sparse approximation with iterative thresholding for audio coding*, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2007.

[YBD08] ———, *Regularized dictionary learning for sparse approximation*, EUSIPCO, 2008.

[YBD09] ———, *Dictionary learning for sparse approximations with the majorization method*, IEEE Trans. on Signal Processing **57** (2009), no. 6, 2178–2191.

[YD09] M. Yaghoobi and M. Davies, *Compressible dictionary learning for fast sparse approximation*, IEEE Workshop on Statistical Signal Processing, Aug. 31- Sept. 3 2009.

[YDD10] M. Yaghoobi, L. Daudet, and M. Davies, *Structured and incoherent parametric dictionary design*, sumitted to IEEE International Conference on Acoustics, Speech and Signal Processing, 2010.

[Zal02] C. Zalinescu, *Convex analysis in general vector spaces*, World Scientific, River Edge, NJ, 2002.

[Zan69] W.I. Zangwill, *Nonlinear programming: A unified approach*, Printice-Hall, 1969.

[ZKY07] Z. Zhang, J.T. Kwok, and D.Y. Yeung, *Surrogate maximization/minimization algorithms and extensions*, Machine Learning **69** (2007), no. 1, 1–33.

[ZP01] M. Zibulevsky and B. A. Pearlmutter, *Blind source separation by sparse decomposition in a signal dictionary*, Neural Computation **13** (2001), no. 4, 863–882.