THE UNIVERSITY *of* EDINBURGH

Metric Learning on High-Dimensional Data with Optimal Transport Distances

Patric-Manuel Fulop

Master of Philosophy

THE UNIVERSITY OF EDINBURGH

2023

To my extended family & friends

Abstract

Optimal transport distances (OT), also known as Kantorovich or Wasserstein distances and its approximate variants such as Sinkhorn Divergences have been widely used in recent years in the field of Machine Learning and its applications. From being used as loss functions in generative model setups such as GANs & unsupervised domain adaptation, to the more recent cluster assignment in self-supervised large state-of-art models such as SWAV, they offer a principled way to compare probability distributions. It is an automatic machinery that takes as an input a ground metric on the data features and lifts this to distances between probabilities on that data space. One of the pitfalls of other often-used methods such as Kullback-Leibler from the family of f-Divergences is the breaking of euclidean metrics in high dimensional spaces, as well as infite solutions when the support of the distributions doesn't match.

In the first part of the thesis, we provide an introduction to Optimal Transport theory, followed by the relevant literature in metric learning & generative modelling that covers OT, including a few advancements in approximations of OT distances, i.e. Sinkhorn divergences, that make training generative models with the Wasserstein distance, faster and scalable.

Two of the main challenges with using OT distances in practice is the cost of computation when the data lives in high dimension, and the choice of a suitable ground metric. Firstly, we look at the recent work by Paty and Cuturi (2019), which aims specifically at reducing the computational cost by computing OT using low-rank projections of the data, seen as discrete measures. We extend this approach and show that one can approximate OT distances by using more general families of maps provided they are 1-Lipschitz. The best estimate is obtained by maximising OT over the given family. As OT calculations are done after mapping data to a lower dimensional space, our method scales well with the original data dimension and is robust against noise. We demonstrate the idea with neural networks and provide some insights into using these methods for training generative models.

Secondly, we look at learning the ground metric for OT distances in a supervised manner and compare this to traditional metric learning methods such as learning the parameters of Mahalanobis distances on MNIST.

Finally, we consider potential avenues for future research in this area.

Lay Summary

Within the last 10 years, we have seen data & machine learning become one of the most sought after commodities and industries, with world-changing products being developed. With increasing amounts of data available, and data dimensions increasing, challenges associated to making sense of it, increase as well. In order for applications to scale, we need to have robust methods that understand the similarities and dissimilarities in high-dimensional data spaces.

This thesis aims to enable the efficient and robust learning of similarities within high-dimensional data spaces by reducing the complexity of the space. Specifically, we propose a modular framework that allows one to learn such similarities, known as metrics in a principled way, by projecting the data into less complex spaces. Examples would include understanding differences between images or documents.

In the first part we identify the challenges associated to the most common ways of learning on high dimensional unstructured data, and challenges associated to how similarities are computed using current tools. We propose the use of Optimal Transport (OT) distances as a more principled framework for defining a learning objective. We define and utilise some of their properties to develop cheaper and robust methods for approximating OT distances on high dimensional data. We evaluate on synthetic & image data the results of our proposed framework and make use of neural networks as projectors into less complex, low-dimensional data spaces.

In the second part, we combine the use of models that learn from unlabelled unstructured data, with OT distances as learning objective, to prove the efficacy of learning similarities between different points in a synthetic dataset, that is a challenging scenario for other types of learning objectives.

Finally, we investigate the metrics we learn in comparison with traditional approaches for the task of image retrieval and classification. Our evaluation indicates that some of our proposed method offer similar results, at a fraction of the cost.

Acknowledgements

I want to thank my family, friends & my lovely wife, as well as the amazing colleagues at Neurolabs.

To my advisor, Vincent Danos, thank you for your patience and guidance during these last few years. You have thought me the importance of being proactive in life and having a true passion for what one does with their time. Your ability to think abstractly and frame complex problems in a formal and quantitative way, is something I will always strive for.

To my second supervisor, Guido Sanguinetti, I am incredibly grateful for having met you and having had the opportunity to discuss ideas with you. At a time when my PhD was not going in the right direction, you have made all the difference which allowed me to continue this endeavour. Thank you for pushing me and supporting me.

Special thanks to I.B. Magdau for always being there for me, academically as a role model, but most importantly as a friend. This thesis wouldn't exist without your support. To my friends Paul, Remus, Liviu & Ozi, for always supporting me along the way.

The last 7 years have given me the opportunity to learn and expand my skills and perseverance in ways I would have never thought previously possible, and have given me the right tools which I am grateful to say I am using to expand my ideas and innovate.

Finally, my work wouldn't have been possible with the support of Microsoft Research through its PhD Scholarship Programme and the University of Edinburgh, for which I am thankful.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Patric-Manuel Fulop

Publications

The following publications have been composed during the course of this masters:

Fulop, P.M. and Danos, V., 2021. Efficient estimates of optimal transport via low-dimensional embeddings. NeurIPS Optimal Transport in Machine Learning Workshop. (P. M. Fulop & Danos, 2021)

Pop, R. and **Fulop, P.**, 2018. Deep ensemble bayesian active learning: Addressing the mode collapse issue in monte carlo dropout via ensembles. NeurIPS Bayesian Deep Learning Workshop. (Pop & Fulop, 2018)

Fulop, P., Manataki, A., Agachi, A., Capital, E. and Pop, P., 2019. PREDICTING SURVIVAL AFTER SURGERY FOR BRAIN. ICLR AI for Social Good Workshop. (P. Fulop, Manataki, Agachi, Capital, & Pop, n.d.)

Contents

Figures and Tables

Figures

Tables

Nomenclature

$f_{\#}$	Pushforward of f
$\mathscr{X}$	Set of high-dimensional data points
$\mathscr{X}, d_{\mathscr{X}}$	High-Dimensional Metric Space
$\mathscr{Y}$	Set of low-dimensional data points
$\mathscr{Y}, d_{\mathscr{Y}}$	Low-Dimensional Metric Space
δ	Dirac measure
$\Gamma(\mu, \nu)$	Space of all joint couplings of type γ
$\gamma(\mu, \nu)$	Joint coupling
$\hat{f}_{\phi}$	Pullback of f
$\mathscr{M}_{\mathscr{X}}$	Space of metrics over $\mathscr{X}$
$\mathscr{M}_{P(\mathscr{X})}$	Space of metrics over probability distributions on $\mathscr{X}$
$\mathscr{N}(0, I)$	The normal distribution with mean 0 and identity covariance
$\mathscr{S}$	Family of maps from $\mathscr{X}$ to $\mathscr{Y}$
$\mathscr{P}(\mathscr{X})$	Space of probability distributions on $\mathscr{X}$
μ, ν	Probability measures in $\mathscr{P}(\mathscr{X}), \mathscr{P}(\mathscr{Y})$
Ω	Mahalanobis metric
Σ_d	Probability Simplex
$d_{\mathscr{X}}$	Distance metric on $\mathscr{X}$
$d_{\mathscr{X}}(x, y)$	Distance between two points in $\mathscr{X}$
$d_{\mathscr{Y}}(x, y)$	Distance between two points in $\mathscr{Y}$
D_M	Mahalanobis metric
f_{ϕ}	map f parametrised by ϕ
H	Entropy
KL	KL Divergence
M	Mutual Information
MMD	Maximum Mean Discrepancy
S_k^2	Subspace Robust Wasserstein distance
SD_{ϕ}^2	Sinkhorn Divergences Projected distance squared
SD_{ε}	Unbiased Sinkhorn Divergence
T	Optimal Transport map
TV	Total Variation
W_2^2	The square of 2-Wasserstein distance
W_p^p	The p-power of p-Wasserstein distance
W_{ε}	The regularised Wasserstein distance
DML	Deep Metric Learning

GAN	Generalised Adversarial Network
GAN	Generalised Adversarial Network
GML	Ground Metric Learning
GPW	Generalised Projected Wasserstein
ITML	Information theoretic metric learning
KNN	K-Nearest Neighbour
LMNN	Large Margin nearest neighbour
OT	Optimal Transport
PCA	Principal component analysis
RBM	Restricted Boltzmann Machine
SD	Sinkhorn Divergences
SRW	Subspace Robust Wasserstein
VAE	Variational Autoencoder

Chapter 1

Introduction

1.1 Preface

Over the last decade, machine learning has grown tremendously and has become a vital field in computer science (Jordan & Mitchell, 2015). With the increasing availability of large datasets and powerful computing resources, machine learning has found applications in all industries, ranging from healthcare, natural & biological sciences, to manufacturing, finance & retail.

One of the most notable developments, has been the emergence of deep learning, which has proven to be highly effective in tasks such as image and speech recognition, natural language processing, and autonomous systems. Although its origins can be traced back to the to the 1980-1990s, the work of Hinton and Salakhutdinov (2006) was instrumental in showing the true potential that these models can have. In image classification tasks for example, the ability of deep convolutional neural networks (CNNs) to deal with complex image data has proved to be unrivalled. Deep CNNs, however, require large amounts of labeled training data to reach their full potential. In part, the success of deep learning is owed to these large amounts of data being available, as well as advances in computational resources, which have enabled researchers to train large neural networks. Broadly speaking, machine learning methods can be broadly categorized into supervised and unsupervised learning, as shown in Figure 1.1.

Supervised Learning

In *supervised learning*, a model learns from labelled examples, by mapping inputs to outputs based on these examples, minimising the error between what the model predicts and the ground truth. This is often done using various loss functions, such as cross-entropy for classification tasks or mean squared error for regression tasks. The goal is to then make predictions on unseen new data.

Unsupervised Learning

Machine learning models in the subfield of *unsupervised learning* (Ghahramani, 2004) are concerned with finding hidden patterns from unlabelled data or semi-labelled data. Finding patterns in data without prior knowledge has been a longstanding hard problem with various different approaches and trends over the years, such as clustering, neural networks, dimensionality reduction such as PCA and, more recently, deep learning approaches such as Variational Autoencoders (Kingma & Welling, 2013) or Generative Adversarial Networks (Goodfellow et al., 2014). A subset of such models are also known under the term of generative models because the problem can often be reduced to the estimation of a probability density over the data. This estimate can then be used to generate new samples from the data. In order to better understand what process generated the data, the process of 'learning' consists of optimizing some function over a high-dimensional data space. One of the first and most powerful generative models was the Restricted Boltzmann Machine (RBM) (Ackley, Hinton, & Sejnowski, 1985; Hinton, 2002), a class of undirected graphical models. In more recent years, architectures such as as VAE & GANs as well as Diffusion Models (Croitoru, Hondru, Ionescu, & Shah, 2023) have made strong headway.

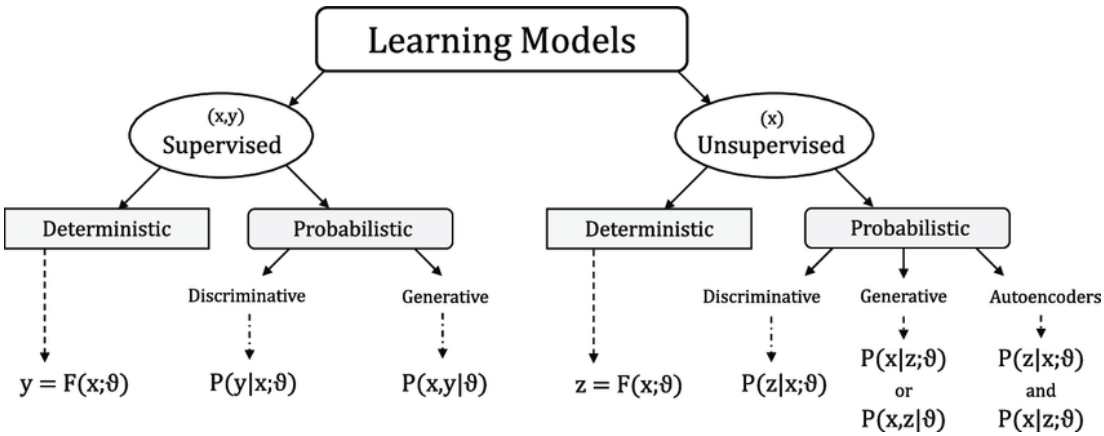


Figure 1.1: Unsupervised vs. supervised learning (Tonello et al., 2019)

In most cases involving unsupervised learning, the commonly used method is to compare the empirical data distribution with a parametrised model distribution using information divergences (f-divergences) such as Kullback-Leibler, reducing the problem to one of comparing probability distributions. On the other hand, understanding density estimation through the lens of optimal transport theory has been a recent endeavour (Feydy et al., 2018; Genevay, Peyré, & Cuturi, 2017; Montavon, Müller, & Cuturi, 2015; Ramdas, Trillos, & Cuturi, 2017). Optimal transport distances (OT) provide a general way of comparing probability distributions using as an input a metric on the data space.

This thesis is about developing frameworks and methods for using optimal transport distances and their approximations, to address some of the challenges arising with unsupervised & supervised learning on high dimensional spaces, such as learning metrics on the data space, or constructing more principled loss functions when training generative models.

1.2 Problem Formulation

The main hypothesis of this thesis is that learning with optimal transport distances can provide more efficient and robust results for tasks such as unsupervised learning of generative models and metric learning on high-dimensional data. Moreover, the use of these distances provides a more principled, performant and robust approach compared to traditional loss functions such as KL Divergence and linear metric learning methods. We develop methods that combine previous approaches in metric learning and optimal transport distances with dimensionality reduction methods and generative modelling approaches, and evaluate on synthetic and real datasets and the task of image retrieval.

Specifically, we tackle the following questions.

1. Can we develop a theoretical framework for approximating OT distances using lower dimensional embeddings? How amenable is this framework to using neural network architectures to embed the data and stochastic optimisation methods, and can we utilise proxy distances in the form of Sinkhorn divergences during optimisation?
2. Can we construct algorithms for OT estimation that preserve the accuracy and are computationally cheaper? Are these algorithms robust to noise and do they create performant metrics or pseudo-metric distances on the data space? How can these algorithms be further used to improve generative models such as GANs?
3. Can we learn the cost function or the ground metric of optimal transport distances using lower dimensional embeddings? How does this metric compare to other metric learning algorithms?

First contributions are presented in Section §3, by introducing General Projection Wasserstein (GPW) distances as approximations to OT distances. We show how we can devise efficient & robust algorithms that can then be further applied in the context of training generative models, to address problems such as mode collapse (Pop & Fulop, 2018).

Secondly, metric learning has been a longstanding idea as outlined in Xing, Jordan, Russell, and Ng (2003), since often enough, high-dimensional data does not have a straightforward metric associated to it such as the euclidean distance. We present the contributions around ground metric learning using low-dimensional projections in Section §4. Our practical approach is to compare different learned metrics on standard synthetic datasets and real ones such as MNIST, using simple algorithms such as k-nearest neighbours.

1.3 Thesis Overview

We divide this thesis into three main components:

- We provide the necessary background to understand concepts such as metric distances and the role they play in machine learning systems, with a particular focus on optimal transport metrics, and their use, as well as their approximations
- We build a generic and efficient method of approximating optimal transport distances by using low-dimensional non-linear spaces and we evaluate their robustness
- We learn and evaluate the ground metric of optimal transport distances on synthetic and real imaging data.

A wide range of machine learning applications, from image recognition and image retrieval (Deselaers, Keysers, & Ney, 2008), to recommender systems and natural language processing, are meant to operate and learn on high dimensional data. Irrespective of the types of model architectures used to learn such systems, it is important to learn robust representations of the samples provided. Sometimes known as 'data embeddings', these have become popular in the last 10 years, since they can be used for multiple downstream tasks, such as domain adaptation (Csurka, 2017), or transfer learning (Weiss, Khoshgoftaar, & Wang, 2016). One of the earliest methods in natural language processing, known as Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) uses deep learning systems to learn word embeddings that preserve semantic similarity. i.e. the data embeddings of the words 'king' and 'queen' are close in the representation space. One can image the myriad of further applications these embeddings can empower, from semantic search, to visual search, as is the case of images, with systems such as Google Lens being prevalent in our lives. We refer to Figure 1.2 for an outline of how Google's Lens product search system works at a high-level. However, most of these systems & methods assume implicitly or explicitly, a definition of *closeness* between embeddings.

Entering the world of similarity metrics and metric distances, these formulations of closeness for data, range from the well-known Euclidean metric and Cosine similarity[1] to more advanced ways to compare embeddings such as Kullback-Leibler and Jensen-Shannon Divergences from the family of f-divergences, all the way to the metrics on the space of probability distributions such as Optimal Transport distances.

However, coming up the with the *right* metric for the problem at hand is often a challenging endeavour. Some of these metrics are known to either not be suitable for high-dimensional data, as is the case for the Euclidean distance or in fact the data lives in low-dimensional manifold where other rules apply. For such situations, the community has developed processes for *metric learning* (Kulis et al., 2013) whereby one can learn an optimal similarity distance between points in space.

1. not all similarity metrics are metric distances, in the mathematical sense

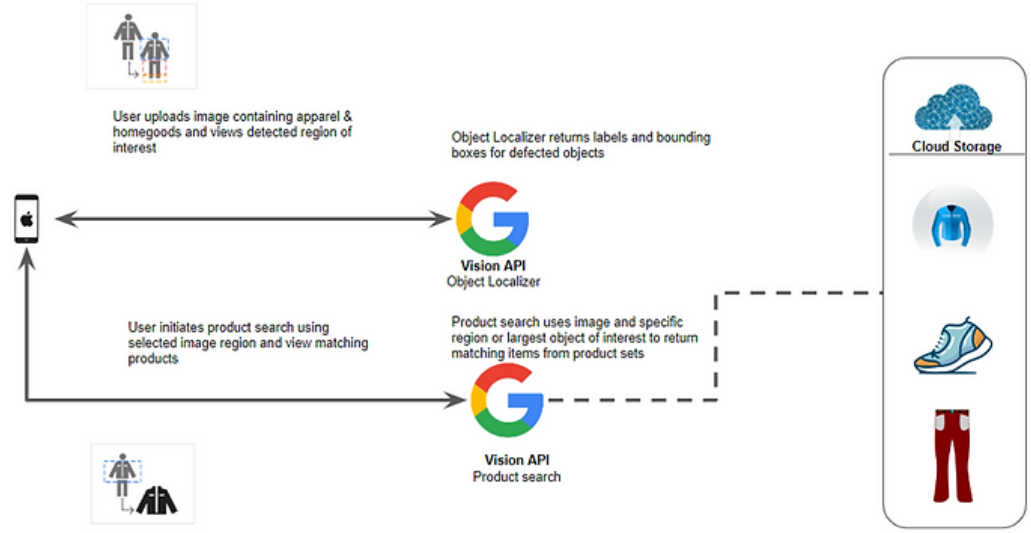


Image Credit: Blog Post on Google Lens

Figure 1.2: Google Lens Visual Search System Diagram

There are other problems when dealing with high-dimensional data and estimating metric distances for them, stemming from something known as the curse of dimensionality, as covered in Appendix A.3. Because increasing the dimensions of space, increases the volume of space, and unless we have exponentially many samples at our disposal, the data then becomes sparse in practice. Algorithms and methods approximating metrics when dealing with sparsity become difficult, with most of the statistical challenges we face highlighted in Johnstone and Titterington (2009).

Although finding the right metric for how to compare points in a high-dimensional space is one of the challenges we face when designing machine learning systems, we are often concerned with finding the right metric between probability distributions. In the context of learning in an unsupervised setting, such as a generative adversarial network, we are often faced with similar challenges, whereby designing the loss functions for such systems is important to ensure that we can model the data distribution properly.

We start in Chapter 2 by providing some historical background on optimal transport and its recent applications in machine learning, such as colour transfer. Next, we introduce mathematical concepts such as metric distances, pullback metric and push-forward of measures, as well as the general p-Wasserstein metric and one of the most used divergence in machine learning, namely, the KL divergence.

We cover the pitfalls of KL in relation to the Wasserstein distance and provide new insights into why using Wasserstein as a loss function is more principled than KL, but also challenges, such as high cost of computation in high-dimensions. We provide the motivation for regularising OT distances to make the computation more efficient and smoother, and trace it back to using the maximum entropy principle. We introduce entropic regularisation and Sinkhorn divergences as smooth approximations of OT with better properties, such as linear scaling of dimension, differentiability and better sample complexity.

In Section §2.2 we give an overview of metric learning approaches using linear methods, but also deep learning ones, and connect them back to using optimal transport for learning metrics. Finally, we conclude in Section §2.3 with an overview of using OT and Sinkhorn Divergences for training large scale generative models such as GANs and VAEs.

In Chapter 3 we cover some of the challenges of computing OT distances between high-dimensional distributions and choosing the correct ground metric, and we introduce the p-Wasserstein distance together with the concept of 'Naturality' of Wasserstein distances, whereby we quantify the relationship between the distances on an ambient space $\mathscr{X}$ and that of a transformed space low-dimensional space $\mathscr{Y}$ through the map T. We provide properties for the map T under which one can successfully approximate the high dimensional OT distances using the lower dimensional ones. We further define a metric with this framework, namely the Generalised Projection Wasserstein (GPW), the supremum of a family of pseudo-metrics obtained by applying OT to low-dimensional projections. We show that the Subspace Robust Wasserstein distance (Paty & Cuturi, 2019) can be seen as an instance of this framework where T is a linear map.

In Section §3.4 we describe an algorithm using Sinkhorn Divergences that provides a robust approximation to OT and computationally, scales linearly with dimension, considerably cheaper than its SRW counterpart. Finally, in Section §3.4.3, we highlight an application of using GPW and Sinkhorn to create a generative model that is able to learn a mixture of Gaussians, an often challenging setup for generative networks, due to the mode collapse problem.

We conclude with Chapter 4 where we introduce a supervised algorithm for learning the ground metric of OT distances in a multiscale fashion, by using neural network maps to embed the data such as Variational Autoencoders and linear maps such as PCA.

We briefly introduce discrete optimal transport, followed by Section §4.2.2, highlighting the original method for learning ground metrics by Cuturi and Avis (2014) (GML). We further showcase in Section §4.4 the impact on convergence of the GML algorithm of finding a better initialiser in the multiclass and binary classification setup. We compare the results of these

methods with a traditional metric learning approach, LMNN (Weinberger & Saul, 2009), by formulating the problem as a retrieval task. We provide an extensive analysis of how K-Nearest Neighbour performance changes as we increase the number of neighbours both during the training phase, as well as at retrieval.

1.4 Contributions

The key contributions of this thesis are:

- introducing the concept of naturality and a novel framework for approximating optimal transport (OT) distances using low-dimensional projections. The framework adapts to both linear and non-linear parametrisable maps, provided they are 1-Lipschitz.
- an efficient and robust method to approximate OT distances in high-dimensions using neural networks and Sinkhorn Divergences, that display low sensitivity of computational costs to the data dimension.
- a method for training generative models using Sinkhorn Divergences that learns a ground cost for OT distances. We provide a study on a multi-modal synthetic dataset for improving mode collapse in GANs using said method.
- a multiscale approach extension to Cuturi and Avis (2014) for learning the ground metric of optimal transport distances by using probabilistic or deterministic maps and the pullback metric through those maps. We provide a comparison study when using PCA & Variational Autoencoders and compare to traditional metric learning approaches such as LMNN.

Chapter 2

Background

2.1 Optimal Transport in Machine Learning

The concept of Optimal Transport can be traced back to Gaspard Monge, the French mathematician who at the end of the 18th century, having been involved in designing multiple defensive projects for fortifications, was pre-occupied with the most efficient way to transport construction materials from one site to another (Peyré, Cuturi, et al., 2017). In its most basic form, we can formalize the problem as a transportation between 1-D histograms, given a cost of transport $d(x,y)$, and assuming histograms are piles of dirt or sand, as depicted in Figure 2.1. The optimal transport plan between $P(x)$ and $Q(y)$ is known as the optimal transport map. It wasn't until the 20th century that *Leonid Kantorovich* formalized and solved the problem of finding the optimal transport plan Kantorovich (1960). He is considered by many to be the father of the field of operations research and in the 1975 was awarded the Nobel Prize for economics for his work on optimal allocation of resources and its implications in economic planning.

The work of George Dantzig on linear programming in the 1950s Dantzig (1963) proposed an efficient numerical scheme to solve a class of convex problems with linear constraints for which finding the optimal transport plan is a special case. Kantorovich or Wasserstein distances, as they are known, are a way to take a distance onto the set $\mathscr{X}$ containing some objects, such as points in a vector space, pixel intensities in images, word embeddings, and transform them into a distance over probabilities on $\mathscr{X}$, i.e. images, documents. For example, consider the one dimensional problem of comparing two histograms of images that denote the intensity levels of grey in a slightly different versions of that image. One can imagine trying to map the darker image onto the lighter one by moving the points in one histogram to match the other (see Santambrogio (2015, §2.5)).

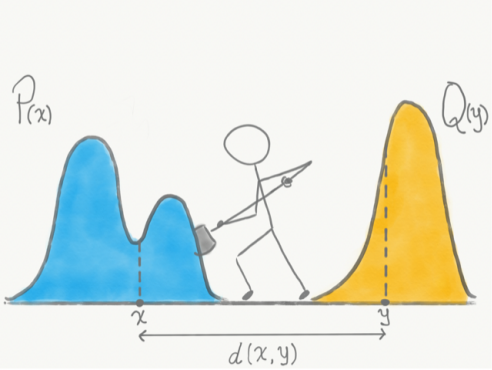


Image Credit: David-Alvarez Melis, Microsoft

Figure 2.1: Optimal Transport between 2 1-D histograms

In the field of machine learning & deep learning, optimal transport has caught on with the community in the last 10 years, with the introduction of the regularised Wasserstein distances by Cuturi (2013) and due to its well-defined theoretical properties when used to compare probability distributions, such as learning a model distribution by fitting a data distribution to it. In order to develop an intuition behind optimal transport and its potential applications in machine learning, we provide an example below of how it can be used. Colour transfer aims to map a color palette from a target image (represented as a the 3-channel RGB histogram or point cloud in $[0,1]^3$) to a source image. In Figure 2.2 we can see the two point clouds between which we can use optimal transport to devise an algorithm to effectively move the colours from target to source. Similar to a traditional gradient based method, for each iteration, we move in the direction of the gradient of the optimal transport distance w.r.t to the source image. In Figure 2.3 we observe the results of the algorithm on an high definition image after 10 iterations and 20 iterations of moving in the direction of such gradient. We can see that even with the difficulty of the terrain in the source image, the algorithm is able to sharply map the colour from the target image. Other areas such as domain adaptation, which aims to find labels for a target dataset by transferring the knowledge from the source, equally benefit from such approaches. However, the optimisation methods needed to produce such algorithms only became efficient in the last few years, with one of the major milestones being the introduction of Sinkhorn Divergences (Feydy et al., 2018; Genevay et al., 2017), as unbiased estimates for OT. In the next sections, we provide the necessary background and notions needed to tackle Chapter 3 and Chapter 4.

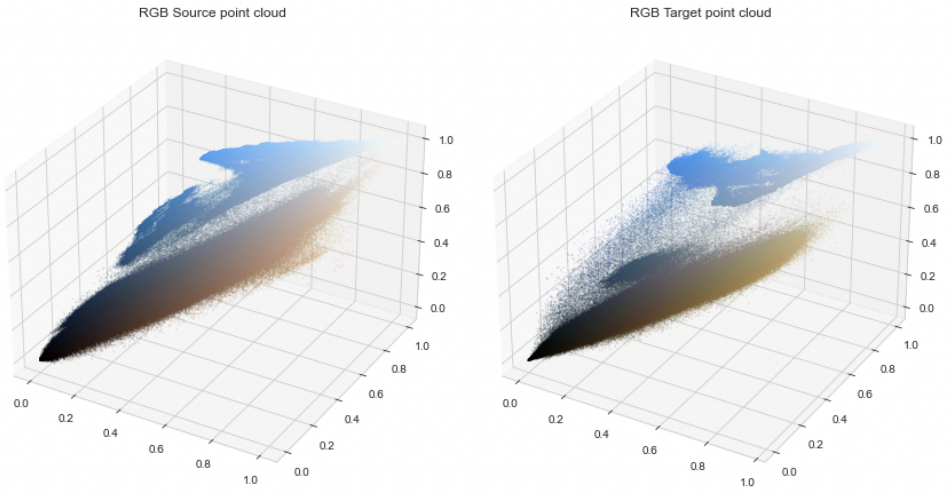


Figure 2.2: RGB Samples for 2 Images in $[0,1]^3$

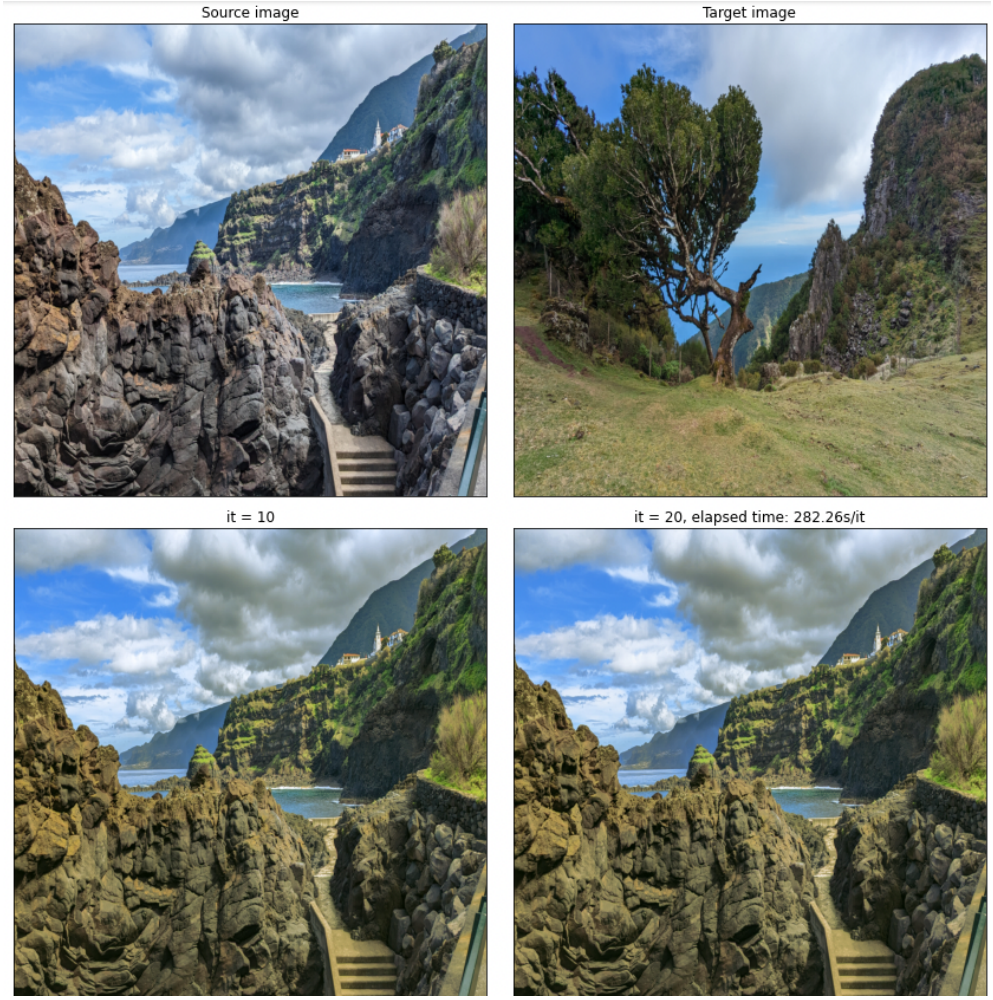


Figure 2.3: Colour Transfer using Sinkhorn Divergences.
Top: Source Image (Left) and Target Image (right)
Bottom: Source Image with colour transferred after 10 and 20 optimisation steps

2.1.1 Metric Distances & Notions

We start with a brief reminder of the basic notions needed. Let $\mathscr{X}$ be a set equipped with a map $d_{\mathscr{X}} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_{\geq 0}$ with non-negative real values. The pair $(\mathscr{X}, d_{\mathscr{X}})$ is said to be a metric space and $d_{\mathscr{X}}$ is said to be a metric on $\mathscr{X}$ if it satisfies the usual properties of positivity, symmetry & triangle inequality:

- $d_{\mathscr{X}}(x,y) = 0$ if and only if $x = y$
- $d_{\mathscr{X}}(x,y) = d_{\mathscr{X}}(y,x)$
- $d_{\mathscr{X}}(x,z) \leq d_{\mathscr{X}}(x,y) + d_{\mathscr{X}}(y,z)$

If $d_{\mathscr{X}}$ verifies the above, except for the first condition, it is called a *pseudo-metric* and $(\mathscr{X}, d_{\mathscr{X}})$ is said to be a pseudo-metric space. For a pseudo-metric, it may be that $d_{\mathscr{X}}(x,y) = 0$ while $x \neq y$. We write $d_{\mathscr{X}} \leq d'_{\mathscr{X}}$ if for all x, y and $d_{\mathscr{X}}(x,y) \leq d'_{\mathscr{X}}(x,y)$. It is easy to see that the following holds true:

1. "$\leq$" is a partial order on pseudo-metrics over X
2. "$\leq$" induces a complete lattice structure on the set of pseudo-metrics over $\mathscr{X}$
3. suprema are computed pointwise, but not infima.

Next, consider $\mathscr{X}$, $\mathscr{Y}$, two metric spaces equipped with respective metrics $d_{\mathscr{X}}$, $d_{\mathscr{Y}}$. A map f from $\mathscr{X}$ to $\mathscr{Y}$ is said to be α-Lipschitz continuous if $d_{\mathscr{Y}}(f(x), f(x')) \leq \alpha d_{\mathscr{X}}(x,x')$. A 1-Lipschitz map is also called *non-expansive*, whereas for $\alpha < 1$, the map f is *contractive*. In case of equality for the above equation, f is known as an *isometry*. An example of such a map would be rotations in the euclidean plane, where distances are preserved.

Pullback

Given a map f from $\mathscr{X}$ to $\mathscr{Y}$ one defines the *pullback* of $d_{\mathscr{Y}}$ along f as:

$$\hat{f}(d_{\mathscr{Y}})(x,x') = d_{\mathscr{Y}}(f(x), f(x')) \tag{2.1}$$

We will also refer to this as *pullback metric*. It is easily seen that:

1. $\hat{f}(d_{\mathscr{Y}})$ is a pseudo-metric on $\mathscr{X}$.
2. $\hat{f}(d_{\mathscr{Y}})$ is a metric if and only if f is injective.
3. $\hat{f}(d_{\mathscr{Y}}) \leq d_{\mathscr{X}}$ if and only if f is non-expansive.
4. $\hat{f}(d_{\mathscr{Y}})$ is the least pseudo-metric on the set $\mathscr{X}$ such that f is non-expansive from $(\mathscr{X}, \hat{f}(d_{\mathscr{Y}}))$ to $(\mathscr{X}, d_{\mathscr{Y}})$

Thereafter, we assume that all metric spaces considered are complete and separable, i.e. have a dense countable subset.

Let $(\mathscr{X}, d_{\mathscr{X}})$ be a (complete separable) metric space. Let Σ_X be the σ-algebra generated by the open sets of $\mathscr{X}$, also known as the the Borelian subsets. We write $\mathscr{P}(\mathscr{X})$ for the set of probability distributions on $(\mathscr{X}, \Sigma_X)$.

Push-forward

Given a measurable map $f : \mathscr{X} \to \mathscr{Y}$, and $\mu \in \mathscr{P}(X)$ one defines the *push-forward* of μ along f as:

$$f_{\#}(\mu)(B) = \mu(f^{-1}(B)) \tag{2.2}$$

for $B \in \Sigma_Y$. It is easily seen that $f_{\#}(\mu)$ is a probability measure on $(\mathscr{Y}, \Sigma_Y)$

Given μ in $\mathscr{P}(\mathscr{X})$, ν in $\mathscr{P}(\mathscr{Y})$, a coupling of μ and ν is a probability measure γ over $\mathscr{X} \times \mathscr{Y}$ such that for all A in Σ_X, B in Σ_Y, $\gamma(A \times \mathscr{X}) = \mu(A)$, and $\gamma(\mathscr{X} \times B) = \nu(B)$. Equivalently, $\mu = \pi_{0\#}(\pi)$, and $\nu = \pi_{1\#}(\pi)$ for π_0, π_1 the respective projections. We also define the probability simplex $\Sigma_d = \{u \in \mathbb{R}_+^d | \sum_i^d u_i = 1\}$ We write $\Gamma(\mu, \nu)$ for the set of couplings of μ and ν, or the set of all joint probability distributions of μ and ν.

Finally, for experiments carried out, we deal with the space $(\mathscr{X}, d_{\mathscr{X}})$ the real positive space of dimension d, equipped with the $L1$ or $L2$ norms, $d_{\mathscr{X}} = L1, L2$. In the context of generative models, for a reference probability distribution Z[1] on $\mathscr{Y}$ with a continuous map $g : \mathscr{Y} \to \mathscr{X}$ and push-forward operator $g_\# : \mathscr{P}(\mathscr{Y}) \to \mathscr{P}(\mathscr{X})$, $g_\# \xi = \mu$ we can sample from $\mathscr{X}$ through the map g, i.e. $x = g(z)$.

For all theoretical explanations and equations, we will either use these notations, or we will closely follow the ones outlined in Peyré et al. (2017).

Wasserstein Distances & KL Divergences

The problem of finding an optimal transport map $T : \mathscr{X} \to \mathscr{Y}$, can we traced back to the original Monge problem, between two measures μ, ν on $\mathscr{X}$, with a cost $c : \mathscr{X} \times \mathscr{X} -> \mathbb{R}$, $c = |T(x) - x|$ such that T is given by:

$$\min_{T_\# \mu = \nu} \int c(x, T(x)) \nu(dx) \tag{2.3}$$

This can be equivalently expressed as finding the shortest path between points going from one space to the other, according to a joint probability between the points. The transport map need not be unique. The natural way to do this is to find the cost of transporting points from one distribution to the other by minimizing the average displacement. Minimizing the average will give you an optimal transport plan. We now define the *Wasserstein distance* in equation 3.1, which invariably corresponds to finding an optimal joint coupling of μ and ν, namely $\gamma \in \Gamma(\mu, \nu)$, the set of all joint probability distributions of μ and ν. μ, ν are also known as the marginals of γ.

$$W_p^p = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathscr{X} \times \mathscr{Y}} d(x,y)^p d\gamma(x,y) \tag{2.4}$$

$$W_2^2 = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathscr{X} \times \mathscr{Y}} \frac{||x-y||^2}{2} d\gamma(x,y) \tag{2.5}$$

We will mostly be concerned with the situations Wasserstein-1 or Wasserstein-2 with Manhattan or Euclidean distance cost function. We note that $\mathscr{X}$ does not necessarily have the same support as $\mathscr{Y}$.

---

1. Z is usually $\mathscr{N}(0,1)$ for most generative models

In most of the machine learning applications concerning generative models, whether optimal transport is used or not, we encounter μ as the empirical data distribution and ν_θ as a parametrised model distribution that needs to be learnt from the data. To that extent, we note that the total variation (TV) distance is a particular case of Wasserstein for discrete spaces using the trivial metric as a parameter[2]. The total variation distance for a finite space is then:

$$\begin{aligned}\delta(\mu,\nu_\theta) &= W(e)(\mu,\nu_\theta)\\ &= \sup_{A\in\Sigma}|\mu_d(A)-\nu_\theta(A)|\\ &= 1/2\sum_{A\in\Sigma}|\mu_d(A)-\nu_\theta(A)|\end{aligned} \tag{2.6}$$

Another well known divergence used for measuring distances between distributions is the Kullback-Leibler (KL) divergence, part of the family of f-divergences. Due to its asymmetrical properties and to the fact that maximum likelihood (MLE) is equivalent to minimising the distance between the data and model distributions, it is used throughout the machine learning field as a loss function, for example for training generative models (see Appendix A.2 for a more detailed analysis on the relationship between MLE and KL). Considering μ, the data distribution and ν, the model distribution, we have:

$$KL(\mu||\nu) = \int log\left(\frac{\mu(x)}{\nu(x)}\right)\mu(x)dx \tag{2.7}$$

Wasserstein Dual Formulation

Finding the Wasserstein distance amounts to solving a linear programming problem and in its standard form presented in equation 3.1 is known as the *primal problem (KP)*. One viable way of minimizing a linear function with linear constraints is the simplex method and can be traced back to George Dantzig (Dantzig, 2016) and Von Neumann (Von Neumann & Morgenstern, 2007). The primal problem is finding the minimum cost of transporting all the points from a set X to the points in set Y. For instance, if we are transporting apples from a storage facility, one must consider the distribution induced by the amount of apples sent at a particular time during the day. We then distribute these across the town at shops located at a particular distance from the storage. The transport plan is the amount of apples transported from one point to another, such that cost is minimum. The usual primal formulation can be expressed using the *dual problem (DP)* as was explained in Bertsimas and Tsitsiklis (Bertsimas & Tsitsiklis, 1997) or in Santambrogio (Santambrogio, 2015, §1.6.2). Alternatively, imagine a business person who has α as the costs of buying apples from storage and β as the price they get from selling it to the shop. Their goal is to maximize profit while keeping the total cost lower than the

2. everything that is equal is at distance 0, whereas all that is different distance 1

transportation cost. We provide a definition below and sketch a proof in Appendix A.1.

$$(DP): \max_{\alpha(x),\beta(y)} \int_{\mathscr{X}} \alpha(x) d\mu + \int_{\mathscr{X}} \beta(y) d\nu \tag{2.8}$$

α, β have finite integrals wr.t. measures μ, ν and $\alpha(x) + \beta(y) \leq d_{\mathscr{X}}(x,y)$. As we shall see in Section 2.1.3, the dual formulation is fundamental to understanding the regularised version of the Wasserstein distance.

2.1.2 Wasserstein over Kullback-Leibler

In their seminal paper, Bassetti, Bodini, and Regazzini (2006) introduced the Minimum Kantorovich estimator[3] comparing the empirical distribution, i.e. data distribution, to statistical model distributions, as one often encounters in machine learning. In such situations, one of the issues of computing Wasserstein distances is the complexity of linear programs that scales cubically with the dimension of the sample space, due to the curse of dimensionality, covered by Friedman, Hastie, and Tibshirani (2001). We cover this further in Appendix A.3. This is also known as sample complexity and in the case of using OT as a loss function for learning generative models, it refers to the convergence rate of the loss evaluated on empirical samples of the distribution to the loss evaluated on the true distributions Genevay, Chizat, Bach, Cuturi, and Peyré (2019). Sample complexity for approximating the original distances grows exponentially in dimension for the original distance, as we need more data to represent the sample space accurately.

Although both KL and OT have associated challenges with high-dimensional data spaces, KL and MLE exhibit some specific properties that make them less amenable to be used as density estimators (see Genevay et al. (2017, §2)). Intuitively, KL can be treated as how much information, in bits, we are expected to lose (on average) when we are trying to approximate two distributions. We cover the main challenges below:

1. MLE will have many local maxima for a high dimensional space since we have a wealth of combinations for a given feature space and our samples will not be able to capture all of them.
2. The samples might live on lower dimensional manifolds and MLE will be zero for non-observed points.
3. Comparing two distributions that might not have common support, would result in the results for MLE being undefined. Wasserstein is able to get around this and compare distributions on different supports. We give an indepth example of this challenge in Appendix A.4.

3. we will to the Kantorovich distance as the Wasserstein distance to avoid confusion

Although the idea of replacing various f-divergences such as KL with Wasserstein distances might provide better generalizations, generative models RBMs, VAEs or GANs (Arjovsky, Chintala, & Bottou, 2017; Genevay et al., 2019; Montavon et al., 2015) provide new practical insights. The more recent treatment of the GAN model by Arjovsky et al. (2017) showed empirically that Wasserstein distances can provide more stable training and deal with mode collapse. In Bousquet, Gelly, Tolstikhin, Simon-Gabriel, and Schoelkopf (2017), authors give an overview of the Wasserstein Autoencoder and Genevay et al. (2017) presents an algorithm for training large scale generative models by making use of automatic differentiation for the Sinkhorn method. Furthermore, in Genevay, Cuturi, Peyré, and Bach (2016), the Sinkhorn method is replaced with a stochastic optimization scheme, which is known to convergence faster. Lastly and most recently, Altschuler, Weed, and Rigollet (2017); Schmitzer (2019) provided near-linear time approximation algorithm that performs better than Sinkhorn.

Computing OT distances efficiently and solving some of the challenges around scalability, were brought to the center of attention of the practical machine learning community by the application of the Sinkhorn algorithm (Cuturi, 2013) that solves an approximate problem using a simple matrix scaling algorithm with an entropy regularized objective of the primal problem (KP). They prove this approximation is differentiable, convex and can be efficiently parallelised. He reports various benchmarks on the MNIST dataset, for different ground metrics, and concludes they are all several orders of magnitude faster than computing the original linear program. Lastly, Carlier, Duval, Peyré, and Schmitzer (2017) provides convergence proofs for regularized transport in the limit of vanishing entropy. Below, we provide a short intuitive explanation for why Sinkhorn was revolutionary for computing OT distances.

Following the Maximum Entropy principle covered in Cover and Thomas (2012, §2) and Appendix A.6, we have that the relationship between the entropy of a joint probability distribution and it's marginals is $H(\gamma) \leq H(\mu) + H(\nu)$. For equality, one recovers the maximal entropy for the joint probability and that μ and ν are independent. We also define the mutual information in Appendix A.5, $M(\mu,\nu) = H(\mu) + H(\nu) - H(\gamma) = KL(\gamma||\mu\nu)$ using the KL divergence between the joint and product probability. Furthermore, a constraint is applied, such that the KL divergence of the joint probability γ to $\mu\nu$, is below a threshold $KL(\gamma||\mu\nu) \leq \varepsilon$.

$$H(\gamma) - \varepsilon \leq H(\mu) + H(\nu) - \varepsilon \leq H(\gamma) \tag{2.9}$$

This equates to seeking the joint probability that has a small enough mutual information or one that lies somewhere ε away from the maximal entropy, $H(\gamma) \geq H(\mu) + H(\nu) - \varepsilon$. One can see that for $\varepsilon \to 0$ we have that the coupling is the one with maximum entropy. Increasing ε will get us a *smoother* solution, namely the original OT solution. The next section makes use of this result to introduce the regularised Wasserstein distance.

2.1.3 Entropic regularisation & Sinkhorn Divergences

Entropic regularisation as defined by Cuturi (2013) relaxes the Wasserstein distance, and is originally solved with the Sinkhorn-Knopp matrix algorithm. Because the original linear problem of finding an optimal coupling is convex and will converge to some minimum with complexity $O(d^3 logd)$, the authors propose to solve a smoothed Wasserstein distance where they penalize it with the entropy of the coupling, which stabilizes the computation and turns it into a convex problem amenable to gradient based methods. One can understand solving (3.1) as obtaining the coupling with a very low entropy, *the lowest entropy*. Thus, solving a minimization problem that penalizes such small entropies will give us a coupling that is more general. This can be seen as an approximate version of optimal transport that won't overfit the data. Authors in Cuturi and Doucet (2014) showed how to solve equation 2.10 for the primal problem using Lagrange multipliers and solved the smoothed dual formulation for Wasserstein barrycenters. We give a brief description of the steps involved for W_1 in Appendix A.7 and present below the smoothed regularised Wasserstein distance:

$$\begin{aligned} W_\varepsilon(\mu,\nu) &= \min_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathscr{X}\times\mathscr{X}} D(x,y)d\gamma(x,y) - \frac{1}{\varepsilon}\mathsf{H}(\gamma(x,y)) \\ W_\varepsilon(\mu,\nu) &= \min_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathscr{X}\times\mathscr{X}} D(x,y)d\gamma(x,y) - \frac{1}{\varepsilon}\mathsf{KL}(\gamma|\mu\otimes\nu) \end{aligned} \tag{2.10}$$

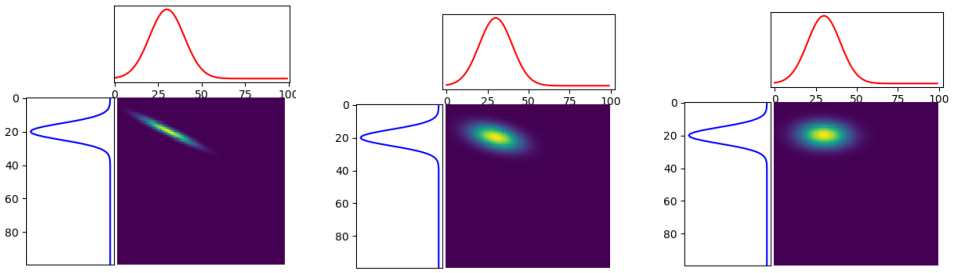
We can look at the effect of ε on equation 2.10 for the case of optimal transport between two Gaussians $\mathscr{N}(20,5)$ and $\mathscr{N}(30,10)$ as seen in Figure 2.4.

When $\varepsilon \to \infty$:

- You recover the linear problem, the optimal Wasserstein coupling. You give no importance to the entropy of the couplings.
- Decreasing ε will penalise distributions that are close, penalizing couplings with lower entropy.

When $\varepsilon \to 0$:

- You recover the coupling with maximal entropy, i.e. the regularization term shadows the Wasserstein objective completely. Potentially, you could see the decrease in ε as a way of changing the learning rate as you get closer and closer to your results.
- From a clear solution for Wasserstein, you move to a more blurry solution (smooth Wasserstein) as you decrease ε.



(a) $\varepsilon = 1000$ **(b)** $\varepsilon = 50$ **(c)** $\varepsilon = 5$

Figure 2.4: Effect of decreasing ε

Sinkhorn Divergences

There have been a number of recent algorithms developed that are more stable than the original developed by Cuturi (2013), such as the scaling introduced by Altschuler et al. (2017); Schmitzer (2016). This methods uses 'detach' during the fixed point iteration to have a spee-dup of the autograd method as well as adding a scale parameter for Sinkhorn which is a special case of simulated annealing, whereby you decrease ε at each iteration letting potentials (dual) adjust in a "coarse-to-fine" fashion. Whilst the original approach is not suitable for continuous measures, the work of Seguy et al. (2017) & Genevay et al. (2016) is one the the first to solve the regularized optimal transport plan using stochastic optimization methods. Both approaches use the dual formulation of the OT distance, but the former uses neural networks to parametrise the dual variables and provides empirical evidence of scalability for their method when a large number of samples is involved as well as better convergence as the size of the samples (dimensions) increases (Seguy et al., 2017, §3.1). The work of Abid and Gower (2018) develops stochastic algorithms for solving a more general class of entropy-regularized OT problems and proves the convergence of a family of approximations that encompass the well-known Greenkhorn and Sinkhorn. However, the most complete approach is defined by Feydy et al. (2018) who defines the family of *Sinkhorn Divergences* that interpolates between OT distances and the more well known MMD distances. One of the core features, is that this divergence is 'unbiased', since it doesn't optimise only for one measure, which allows them to further prove convergence in law.

$$SD_\varepsilon(\mu,\nu) = W_\varepsilon(\mu,\nu) - \frac{1}{2}W_\varepsilon(\mu,\mu) - \frac{1}{2}W_\varepsilon(\nu,\nu) \tag{2.11}$$

Authors prove that there are certain theoretical guarantees, such as convexity and positivity as well as symmetry that hold true, as well as provide a bound for the approximation error between SD_ε and W. All of these properties, make SD_ε amenable to being used as a loss function for learning from empirical data. Most importantly, authors are able to construct an effective numerical scheme for computing the gradients on GPU, scaling up to millions of samples. They also show that the sample complexity is dependent only on $\frac{1}{\sqrt{n}}$ and the

regularization factor ε. This builds on the results from Smola, Gretton, and Borgwardt (2006), where authors prove MMD has sample complexity $\frac{1}{\sqrt{n}}$ and the fact that you can rewrite the regularized OT using the Lagrangian formulation as an expectation over the product measure.

2.2 Metric Learning

One of the most fundamental mathematical concepts used in machine learning is that of a distance metric (See Section §2.1.1) which aims to capture the pairwise-similarity between data points. Distance functions such as the L_1 and L_2 norms are ubiquitous and indispensable for methods such dimensionality reduction, clustering, or classification and evaluation of algorithms. Often though, it is hard to define a distance function that accurately captures similarities in a generic dataset, and that allows one to improve the learning methods mentioned above. It is even more difficult to define such a distance function as the dimensionality of data increases. This prompted the development of machine learning methods that aim to find similarity metrics dependent on the dataset given. Building on the pioneering work of Xing et al. (Xing et al., 2003) who first defined metric learning as a convex optimization problem in the context of clustering, there have been numerous successful algorithms developed since (Davis, Kulis, Jain, Sra, & Dhillon, 2007)Weinberger and Saul (2009) Cuturi and Avis (2014); Lin, Fan, Ho, Cuturi, and Jordan (2020); Muzellec and Cuturi (2019); Paty and Cuturi (2019). We highlight some of the key metric learning methods, with a stronger focus on more recent ones that are connected to optimal transport. Two of the most well known methods in metric learning, Information Theoretic Metric Learning (ITML) (Davis et al., 2007) Large Margin Nearest Neighbour (LMNN) (Weinberger & Saul, 2009) both learn the Mahalanobis distance. We first provide a definition:

Definition 1. ***Mahalanobis distances***
For $L : \mathscr{X} \to \mathscr{X}$ a linear transform on $\mathscr{X}$, denote $M = L^T L \in \mathscr{M}$, a symmetric, positive semi-definite matrix. This is to ensure the squared Mahalanobis distance is itself a pseudo-metric in $\mathscr{M}$.

$$D_M(x,x') = (x-x')^T M(x-x') \tag{2.12}$$

Intuitively we observe that D is trying to capture relations within $\mathscr{X}$ through M. One notices that for $D = I$, we recover the Euclidean distance. Similarly, Mahalanobis distances appear in the exponent of a multivariate Gaussian where M is taken to be the inverse of the covariance Ω^{-1}. We can see that the problem of Mahalanobis metric learning can be formulated as either learning M or L. One can easily show that computing D is equivalent to performing a transformation through L and then computing the Euclidean distance within the transformed space, i.e. the pullback.

One of the advantages of learning M is that, while L uniquely defines M, M is uniquely defined by L up to a rotation. Furthermore, in case the rank of M is smaller than the dimension of L this allows one to express the Mahalanobis metric in a lower dimensional space, further outlining the attractiveness of learning M. Another advantage is the ability to express it as a constrained convex optimization problem where the goal is to learn M, s.t. $M \in \mathscr{M}$. We now cover the relationships between Mahalanobis, LMNN, ITML and Optimal transport approaches to learning metric distances.

LMNN is a supervised metric learning algorithm where one defines local constraints by imposing that the k-nearest neighbours of any point be in the correct class. This results in a convex program over $M \in \mathscr{M}_+$ where the goal is to minimize $d_{\mathscr{X}}$ between pairs of points in a neighbourhood while keeping away points from a different class (Weinberger & Saul, 2009, eq. 4). The method learns Ω in a supervised fashion by bringing similar examples closer, and pulling apart different ones. This is a similar idea to the convex program developed for ground metric learning using optimal transport in Cuturi and Avis (2014).

ITML, by Davis et al. (2007), aims to minimize the KL divergence between two multivariate Gaussians parametrised by M and M_0, where often $M_0 = I$ (Bellet, Habrard, & Sebban, 2013, eq. 5). Once again, they use supervision and nearest neighbours and add constraints that distances between similar points are kept under a certain threshold, and conversely, dissimilar points are kept large enough, bigger than a big enough threshold.

We also mention that principal component analysis (PCA) can be used as a metric learning method, where we learn the linear transformation L that projects into the space of eigenvectors that maximizes the covariance of the data. However, we mainly use PCA as a dimensionality reduction technique, prior to metric learning, as Weinberger and Saul (2009) take a note that learning the metric on a lower dimensional space, improves the accuracy of a KNN classifier.

Lastly, we mention the impact of deep metric learning algorithms & approaches on the literature. With the advent of deep neural networks, the community moved from traditional approaches covered above, to approaches whereby neural networks are used to create a representative space, and the loss functions chosen are the most important. One of the first approaches for deep metric learning was to use the Contrastive Loss Hadsell, Chopra, and LeCun (2006), which follows a similar approach to LMNN, for bringing similar classes together and separating different ones, but the data is embeded through a neural network and a hyper-parameter is introduced to define a lower bound distance between samples of different classes. The Triplet Loss introduced by Schroff, Kalenichenko, and Philbin (2015) for the purposes of designing more performant face recognition retrieval, introduced an extra parameter known as a 'margin'. The loss is designed in such a way as to maximise the distance between the anchor and the negative samples, and minimise the distance between the anchor and

positive samples. The excellent work by Zhai and Wu (2018) showed that removing the last layer of a standard classification network also gives state-of-the-art results for image retrieval and can be considered a form of deep metric learning, whereas Musgrave, Belongie, and Lim (2020) has created a PyTorch library encompassing all of the above methods and more.

Metric Learning with Optimal Transport

The first time metric learning appears in the Optimal Transport literature is in the work of Cuturi and Avis (2014) where authors learn the cost function, or the ground metric for a discrete optimal transport distance in a supervised way, in similar fashion to LMNN. This will be further covered in Section §4.2.2.

In the original work on Subspace Robust Wasserstein (SRW) distances, Paty and Cuturi (2019) propose a max-min robust variant of computing Wasserstein distance, by projecting onto a k-dimensional subspace. The paper follows a core idea coming from Santambrogio (2015), namely that OT on the real line may be sufficient to extract geometric information from high dimensional data. In this paper, they project the measures on a k-dimensional subspace $k \geq 2$, that maximizes their transport cost, similar to learning a ground metric approach. This can be achieved by minimizing the sum of k largest eigenvalues of the second order moment matrix of the transport cost. Further details on their methodology can be found in Appendix B.1. In Projection Robust Wasserstein distance and Riemannian optimization, Lin et al. (2020) present an extension of the core optimisation algorithm in Paty and Cuturi (2019) that is efficient for high dimensions. The authors prove that solving this max-min problem is equivalent to maximising over the Stiefel manifold (cost function) and minimising over the polytope (optimal plan). In Paty and Cuturi (2020), authors follow a similar idea that we present in Chapter 3 and they show that *regularised optimal transport* is the same as maximizing *non-regularised OT* with respect to the ground cost. Authors in Muzellec and Cuturi (2019) present an interesting approach, similar to the Sliced Wasserstein distances (Kolouri, Pope, Martin, & Rohde, 2018), whereby the projection is in a linear subspace $\mathbb{E}$ of dimension k. They derive some closed form solutions for Gaussian measures as well as present the an application for color transfer.

Although most of the methods reviewed are concerned with learning an actual distance, one of the main areas of interest for optimal transport distances has been as loss functions for learning generative models. The work from Kolouri et al. (2018) on defining the Sliced Wasserstein distance led to a type of Variational Autoencoder defined in Kolouri, Pope, Martin, and Rohde (2019), to complement the similar Sinkhorn Autoencoder by Patrini et al. (2018). Previously, the work of Genevay et al. (2017) was in fact the first one to use the Sinkhorn Divergence as a loss function for an Autoencoder.

Finally, one of the first works to use Wasserstein distances for generative modeling was (Cuturi & Doucet, 2014) in Wasserstein Barycenters, where authors create a generalised approach to clustering, by finding a mean measure. The second one was the Restricted Boltzmann Machines using the Wasserstein distances as a loss function by Montavon et al. (2015). In the next section, we cover some of the newer generative models that will be used in Chapter 3.

2.3 Generative Models & Sinkhorn Divergences

One of the main challenges in machine learning & deep learning with wide implications across a range of downstream tasks, is to learn from data in an unsupervised manner, i.e. learn the distribution of the data. Two of the main innovations in the last 10 years that have arguably changed the field of deep learning, have been encoder-decoder architectures, specifically two generative frameworks, Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN).

VAEs, one of the first encoder-decoder architectures as introduced by Kingma and Welling (2013), are principled approaches towards modelling the distribution of data using a probabilistic latent space, and learn low-dimensional representation of the data distribution, i.e. a latent space that is parametrisable. The architecture consists of an encoder that is able to model a mean and variance using KL divergence as a loss function, in the latent space, as well as decoder that is able to reconstruct the input data accurately. Because of the probabilistic latent space, maximizing the likelihood becomes an intractable problem, as it is equivalent to finding a posterior probability on the data, conditioned on the latent space. For an in-depth review See Appendix A.8 and (Blei, Kucukelbir, & McAuliffe, 2017, §2.2).

GANs, as introduced by Goodfellow et al. (2014) have been applied to numerous tasks, such as image synthesis, style transfer, data augmentation, and unsupervised representation learning. They involve the training of two networks simultaneously, a generator and a discriminator, in an adversarial min-max game. The original formulation uses the KL divergence as a loss function for the *generator*, which learns how to generate accurate samples starting from noise data, whereas the *discriminator* has a binary divergence loss function that distinguishes between real and fake samples generated (see Figure 2.5 for a depiction of such a system). However, training GANs can be challenging due to issues like mode collapse, where the generator gets stuck in a local minima (Pop & Fulop, 2018), vanishing gradients, where the discriminator is too powerful so the generator has null gradient, and non-convergence, with the networks and training regimes not being expressive enough to cover the full sample space of the data distribution. Some of these issues can be traced back to the choice of loss function, and the challenges we've covered in Section 2.1.2. Indeed, the various successful techniques and modifications that have been proposed including Wasserstein GANs (WGANs) (Arjovsky

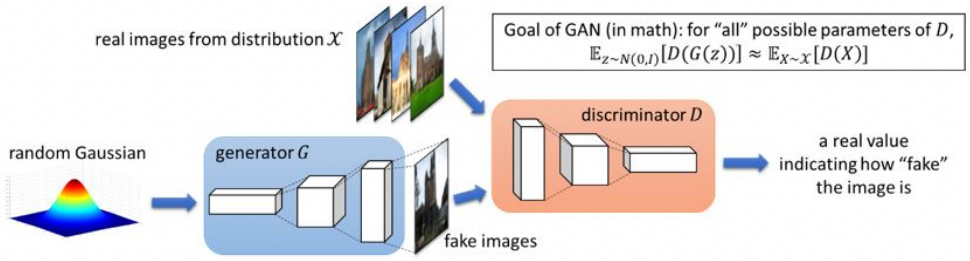


Image Source

Figure 2.5: Generative Adversarial Network

et al., 2017) which uses a Wasserstein-1 proxy distance, MMD-GAN (C.-L. Li, Chang, Cheng, Yang, & Póczos, 2017), VEEGAN (Srivastava, Valkov, Russell, Gutmann, & Sutton, 2017), and spectral normalisation by Miyato, Kataoka, Koyama, and Yoshida (2018) that tackle some of these issues.

One of the main motivations behind using a principled distance metric between distributions for both VAEs and GANs, such as the optimal transport distance, is to be able to learn between distributions with non-overlapping support, but also to be able to specify the ground metric on the data space that we'd like to use. In Genevay et al. (2017) authors present the first approach towards learning a generative model with optimal transport based losses, which is able to overcome some of the challenges in OT, by utilising the regularised unbiased Sinkhorn divergence presented in Section §2.1.3 and automatic differentiation, for computing gradients. Although they provide some insights that these divergences can interpolate between MMD and true OT, Feydy et al. (2018) further expands on the theoretical properties.

We give a brief overview of the training involved and note that maximising the discriminator using SD_ε amounts to learning a cost function that brings similar samples closer. The algorithm is essentially learning a ground metric on the embedding space, such that the latent representations are better separated.

The objective loss for the both the generator and discriminator is SD_ε seen as functions of θ and a neural network parametrised by ϕ.

Writing β for the data seen as a distribution over $\mathbb{R}^d$ and $g_\theta : \mathscr{Z} = \mathbb{R}^k \to \mathbb{R}^d$ for the generating network with $k \ll d$ we can write the model distribution generated through g_θ as the push-forward of some reference Gaussian vector $\zeta = \mathcal{N}(0,1)$ over $\mathscr{Z}$, $\alpha_\theta = g_{\theta\#}\zeta$. While for the discriminator network the goal is to learn a parametrized f_ϕ that maximizes $SD_\varepsilon(f_{\phi\#}\beta, f_{\phi\#}\alpha_\theta)$, the goal for the generator network is to bring α_θ as close as possible to β in Wasserstein space.

The flow of training for this type of construct is the following:

1. Train the discriminator by running stochastic gradient 'ascent' on ϕ and find the best ground metric for the data:

$$d_{\mathscr{P}(\mathscr{X})}(\alpha_\theta, \beta) \quad := \quad \sup_\phi SD_\varepsilon(d_\phi)(f_{\phi\#}\alpha_\theta, f_{\phi\#}\beta)$$

2. Train the generator by running stochastic gradient 'descent' on θ $2-5$ more steps than the discriminator, as this is the more challenging task.
3. At convergence, pick the winning α_θ defining the good generator, and f_ϕ defining the good ground metric.

In Patrini et al. (2018) authors apply the same ideas to construct a likelihood free Autoencoder, namely the Sinkhorn Autoencoder, where the loss function for the encoder is replaced by the Sinkhorn Divergence loss, and show that the latent space is more structured and better aligned with the data manifold. In particular, they showcase on several datasets such as MNIST, CelebA and dSprites that the latent space provides better disentanglement, providing a clearer understanding for the factors of variation in the data. Similarly, Kolouri et al. (2018) provides a similar approach, where authors use Wasserstein-2 projections on the one-dimensional Euclidean space (see (SW) Sliced-Wasserstein Distances in Kolouri et al. (2018)) as regularisation to create a loss function composed of a the classical Wasserstein distance and the computationally cheaper one, SW. In Salimans, Zhang, Radford, and Metaxas (2018) proposes to combine the primal formulation KP from optimal transport with an energy distance defined in an adversarially learned feature space, resulting in a highly discriminative distance function with unbiased mini-batch gradients. In fact, the loss function that he defines, can be also understood as the square root of the unbiased SD_ε. Finally, they train their networks against standard practice, and train the generator more often than the discriminator, such that the cost function doesn't become degenerate. This is something we explored in Section §4.4 as well. Since the discriminator has such a principled loss function, compared to the original fake/real loss function, it will learn a lot faster than the generator, who needs to be trained for more steps. As covered in Feydy et al. (2018), one of the key aspects of using SD_ε to train generative models is that there is a possibility to use mini-batches and SDG to do optimisation. Work by Fatras, Zine, Flamary, Gribonval, and Courty (2019) provides insights into the asymptotic rate of convergence for using OT as a loss function, as well as gradient properties, and shows that it is still an unbiased estimator for the true OT distance. They provide examples of applications to colour transfer. In summary, the new wave of Sinkhorn & Wasserstein distances, offer a more principled approach for training large scale generative models, both in terms of creating better representations of the data, generating samples with more fidelity and learning how to separate samples better through a more stable discriminator training. After the learning process is over, in principle, we have a 'better' f_{ϕ_0} which we can use to compare samples in $\mathbb{R}^d$, i.e. images, by comparing them through f_{ϕ_0}:

$$d_{\mathscr{X}}(x, x') = d_{\phi_0}(f_{\phi_0}(x), f_{\phi_0}(x'))$$

This learned metric could be interesting because it can separate data well, i.e. learn a metric on the data space, and as such has direct consequences for transfer learning, few-shot learning & domain adaptation.

Chapter 3

Efficient Estimates of Sinkhorn Divergences

Optimal transport distances (OT) have been widely used in recent work in Machine Learning as ways to compare probability distributions. These are costly to compute when the data lives in high dimension. Recent work aims specifically at reducing this cost by computing OT using low-rank projections of the data, seen as discrete measures (Paty & Cuturi, 2019). We extend this approach and show that one can approximate OT distances by using more general families of maps provided they are 1-Lipschitz. The best estimate is obtained by maximising OT over the given family. As OT calculations are done after mapping data to a lower dimensional space, our method scales well with the original data dimension. We demonstrate the idea with neural networks. Our approach can be seen as learning a parameterized 'ground' cost function and computing OT distances in a much lower dimensional space. We use Sinkhorn Divergences (SD) to approximate OT distances as they are differentiable and allow for gradient-based optimisation. We illustrate on synthetic data how our technique preserves accuracy and displays a low sensitivity of computational costs to the data dimension.

3.1 Introduction

Optimal Transport metrics (Kantorovich, 1960) or Wasserstein distances, have emerged successfully in the field of machine learning, as outlined in the review by Peyré et al. (2017). They provide machinery to lift distances on $\mathscr{X}$ to distances over probability distributions in $\mathscr{P}(\mathscr{X})$. They have found multiple applications in machine learning: domain adaptation (Courty, Flamary, Habrard, & Rakotomamonjy, 2017), density estimation (Bassetti et al., 2006) and generative networks (Genevay et al., 2017; Patrini et al., 2018). However, it is prohibitively expensive to compute OT between distributions with support in a high-dimensional space and might not even be practically possible as the sample complexity can grow exponentially as shown by Dudley (1969). Similarly, work by Weed, Bach, et al. (2019) showed a theoretical improvement when the support of distributions is found in a low-dimensional space. Furthermore, finding a principled ground metric that one should use is a key ingredient to computing accurate OT distances and is not obvious when using high-dimensional data.

In this chapter, we introduce a general framework for approximating high-dimensional OT using low-dimensional projections f by finding the subspace with the worst OT cost, i.e. the one maximizing the ground cost on the low-dimensional space. By taking a general family of parameterisable f_ϕs that are 1-Lipschitz, we show that our method generates a pseudo-metric and is computationally efficient and robust.

In Section §3.2 & Section §3.3 we provide the relevant literature as well as relevant theoretical background on optimal transport and pseudo-metrics. In Section §3.3.2 we define the theoretical framework for approximating OT distances and show how both linear (Paty & Cuturi, 2019) and non-linear projections can be seen as a special instance of our framework. In Section §3.4.1 we present an efficient algorithm for computing OT distances using Sinkhorn Divergences and f_ϕs that are 1-Lipschitz under the L_2 norm. We present in Section §3.4.2 the numerical experiments illustrating the efficiency and robustness of our method, on high dimensional synthetic data. Finally, we conclude in Section §3.4.3 by presenting an extension of our method applied to learning the discriminator of a GAN, and showcase the results on a multi-modal Gaussian dataset.

Contributions The key contributions of this chapter are:

- introducing a general theoretical framework for approximating optimal transport distances in high-dimensions, by using lower dimensional projections.
- introducing a novel algorithm that uses linear neural networks as non-linear projections and Sinkhorn Divergences for approximating OT distances
- showing the improvements in efficiency and robustness of this algorithm on synthetic data
- showing how the algorithm can be applied in training a generative adversarial network using Sinkhorn Divergences and evaluating on multi-modal data.

3.2 Related Work

One of the earlier ideas from Santambrogio (2015) showed that OT projections in a 1-D space may be sufficient enough to extract geometric information from high dimensional data. This further prompted Kolouri et al. (2018) to use this method to build generative models, namely the Sliced Wasserstein Autoencoder. Following a similar approach Paty and Cuturi (2019) and Muzellec and Cuturi (2019) project the measures into a linear subspace E of low-dimension k that maximizes the transport cost and show how this can be used in applications of color transfer and domain adaptation. This can be seen as an extension to earlier work by Cuturi and Doucet (2014) whereby the cost function is parameterized.

One of the fundamental innovations that made OT appealing to the machine learning community was the seminal paper by Cuturi (2013) that introduced the idea of entropic regular-

ization of OT distances and the Sinkhorn algorithm. Since then, regularized OT has been successfully used as a loss function to construct generative models such as GANs (Genevay et al., 2017) or RBMs (Montavon et al., 2015) and computing Barycenters (Claici, Chien, & Solomon, 2018; Cuturi & Doucet, 2014). More recently, the new class of Sinkhorn Divergences was shown by Feydy et al. (2018) to have good geometric properties, and interpolate between Maximum Mean Discrepancies (MMD) and OT. We refer the reader to Section §2.3 for a more detailed analysis of other generative models using OT as loss functions.

3.3 Methodology

For an introduction to the notions used below and general terminology, we refer the reader back to the background Section §2.1.1. There are several ways to lift a given metric structure on $d_{\mathscr{X}}$ to one on $\mathscr{P}(\mathscr{X})$. We will be specifically interested in metrics on $\mathscr{P}(\mathscr{X})$ derived from optimal transport problems.

The p-Wasserstein metric with $p \in [1,\infty)$ is defined by:

$$W_p(d_{\mathscr{X}})(\mu,\nu)^p = \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathscr{X}\times\mathscr{X}} d^p_{\mathscr{X}}\, d\gamma \tag{3.1}$$

Villani (2008) establishes that if $d_{\mathscr{X}}$ is (pseudo-) metric so is $W_p(d_{\mathscr{X}})$. The natural 'Dirac' embedding of $\mathscr{X}$ into $\mathscr{P}(\mathscr{X})$ is isometric, i.e. there is only one coupling.

The idea behind the definition is that $d^p_{\mathscr{X}}$ is used as a measure of the cost of transporting units of mass in $\mathscr{X}$, while a coupling γ specifies how to transport the μ distribution to the ν one. One can therefore compute the mean transportation cost under γ, and pick the optimal γ.

In most of this chapter, we are concerned with the case $\mathscr{X} = \mathbb{R}^d_+$ for some large d with a metric structure $d_{\mathscr{X}}$ given by the Euclidean norm, and we wish to compute the W_2 metric between distributions with finite support. Since OT metrics are costly to compute in high dimension, to estimate these efficiently, and mitigate the impact of dimension, we will use a well-chosen family of fs to push the data along a map with a low dimensional co-domain $\mathscr{Y}$ also equipped with the Euclidean metric. The reduction maps may be linear or non-linear. However, they have to be non-expansive to guarantee that the associated pullback metrics are always below the Euclidean one, and therefore have a lower bound for $W_2(d_2)$. In the next section, we present the concept of *Naturality*, which we'll further use for Section §3.3.4.

3.3.1 Naturality

For two pseudo-metrics $d_{\mathscr{X}}$ on $\mathscr{X}$ and $d_{\mathscr{Y}}$ on $\mathscr{Y}$ respectively, we denote a probabilistic map $T : \mathscr{P}(\mathscr{X}) \to \mathscr{P}(\mathscr{Y})$, or a deterministic one $f : \mathscr{X} \to \mathscr{Y}$. According to Section 2.1.1 the pullback of the pseudo-metric $d_{\mathscr{Y}}$ through T is:

$$\hat{T}(d_{\mathscr{Y}})(x,x') := d_{\mathscr{Y}}(T(x),T(x')) \tag{3.2}$$

Definition 2. ***Lipschitz continuity***
T is α-Lipschitz continuous if $d_{\mathscr{Y}}(T(x),T(x')) \leq \alpha d_{\mathscr{X}}(x,x')$. For $\alpha < 1$, the map T is a contraction, whereas for $\alpha = 1$ is non-expansive.

Definition 3. ***Isometry***
T is an isometry if $d_{\mathscr{Y}}(T(x),T(x')) = d_{\mathscr{X}}(x,x')$

Lemma 1. *Given $u : \mathscr{X} -> \mathscr{Z}$ with u α-Lipschitz, u factors uniquely through T and conversely, any factored form is Lipschitz by composition and the fact that T is an isometry.*

Proof. By 2, if u is α-Lip:

$$\begin{aligned} d_{\mathscr{Z}}(u(x),u(x')) &\leq \alpha \hat{T}(d_{\mathscr{Y}})(x,x') \\ &= \alpha d_{\mathscr{Y}}(T(x),T(x')) \end{aligned} \tag{3.3}$$

□

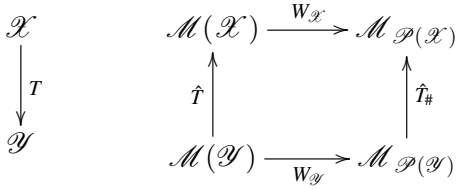
If we define $\hat{T}(d_{\mathscr{Y}})(x,x') := W(d_{\mathscr{Y}})(T(x),T(x'))$, we prove that for an optimal ground metric $d^*_{\mathscr{Y}}$ on $\mathscr{Y}$, $\hat{T}(d^*_{\mathscr{Y}})$ is the best among $\hat{T}(d_{\mathscr{Y}})$.

$$W(d_{\mathscr{Y}})(\tilde{\mu},\tilde{\nu}) = \inf_{\delta \in \Delta(\tilde{\mu},\tilde{\nu})} \int_{\mathscr{Y}\times\mathscr{Y}} d_{\mathscr{Y}}(y,y') d\delta \tag{3.4}$$

$$W(\hat{T}d_{\mathscr{Y}})(\mu,\nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathscr{X}\times\mathscr{X}} d_{\mathscr{Y}}(Tx,Tx') d\gamma \tag{3.5}$$

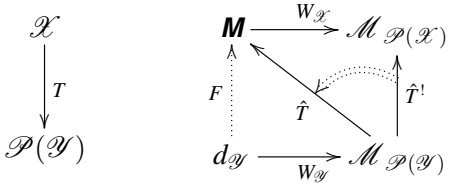
Lemma 2. *For a probabilistic map T and $x,x' \sim \mu,\nu$ as well as $y,y' \sim \tilde{\mu} = T_{\#}\mu, \tilde{\nu} = T_{\#}\nu$, $W(d_{\mathscr{Y}})(\tilde{\mu},\tilde{\nu}) \leq W(\hat{T}d_{\mathscr{Y}})(\mu,\nu)$ holds true, with equality if T is deterministic.*

The second 'equality' statement in 2 is equivalent to the below diagram commuting, which means that the Wasserstein distance W is a natural transformation between the functor $\mathscr{M}$ and $\mathscr{M}_{\mathscr{P}(\mathscr{X})}$. That is a map between the functor, metrics $d_{\mathscr{X}}$ on $\mathscr{X}$, and the functor, metrics on probabilities of $\mathscr{P}(\mathscr{X})$.



(3.6)

The first statement is equivalent to the below diagram. It does not commute anymore, i.e. going from $d_{\mathscr{Y}}$ to $\mathscr{M}_{\mathscr{P}(\mathscr{X})}$ leaves us with a pointwise larger pseudo-metric than taking the other branch. Following the arrow can be thought of in terms of the entropic regularization covered in Section 2.1. In this case we have an 'algebraic' regularization of the other branch.



(3.7)

Proof. First we show that the distance over the transformations of $\mathscr{X}$ is a non-expansive map, i.e. $W(d_{\mathscr{Y}})(\tilde{\mu},\tilde{\nu}) \leq W(\hat{T}d_{\mathscr{Y}})(\mu,\nu)$. Since T is 1-Lipschitz as shown in 1, we use the dual formulation from Appendix A.1 and Villani's proof in Villani (2008, §5.4) that shows cost functions are pseudo-metrics.
□

3.3.2 Approximate OT with General Projections - GPW

With the ingredients from the above section in place, we can now construct a general framework for approximating Wasserstein-like metrics by low-dimensional mappings of $\mathscr{X}$. We write simply W instead of W_p as the value of p plays no role in the development.

Pick two metric spaces $(\mathscr{X},d_{\mathscr{X}}),(\mathscr{Y},d_{\mathscr{Y}})$, and a family $\mathscr{S} = (f_\phi : \mathscr{X} \to \mathscr{Y};\ \phi \in S)$ of mappings from $\mathscr{X}$ to $\mathscr{Y}$. Define a map from $\mathscr{P}(\mathscr{X}) \times \mathscr{P}(\mathscr{X})$ to non-negative reals as follows:

$$d_{\mathscr{S}}(\mu,\nu) = \sup_{\mathscr{S}} W(d_{\mathscr{Y}})(f_{\phi\#}(\mu), f_{\phi\#}(\nu)) \tag{3.8}$$

Equivalently and more concisely $d_{\mathscr{S}}$ can be defined as:

$$d_{\mathscr{S}}(\mu,\nu) = \sup_{\phi} W(\hat{f}_\phi(d_{\mathscr{Y}}))(\mu,\nu) \tag{3.9}$$

It is easily seen that:

1. the two definitions are equivalent
2. $d_{\mathscr{S}}$ is a pseudo-metric on $\mathscr{P}(\mathscr{X})$
3. $d_{\mathscr{S}}$ is a metric, and not just a pseudo one, if the family f_ϕ jointly separates points in $\mathscr{X}$
4. if the f_ϕs are non-expansive from $(\mathscr{X}, d_{\mathscr{X}})$ to $(\mathscr{Y}, d_{\mathscr{Y}})$, then $d_{\mathscr{S}} \leq W(d_{\mathscr{X}})$

In other words, $d_{\mathscr{S}}$ is the least pseudo-metric which makes all $f_{\phi\#}$ non-expansive from $\mathscr{P}(\mathscr{X}) \to \mathscr{P}(\mathscr{Y})$

The second point follows readily from equation 3.9. Each $\hat{f}_\phi(d_{\mathscr{Y}})$ is a pseudo-metric on $\mathscr{X}$ obtained by pulling back $d_{\mathscr{Y}}$ (see Section §2.1.1). Likewise, it follows that $W(\hat{f}_\phi(d_{\mathscr{Y}}))$ on $\mathscr{P}(\mathscr{X})$, is also a pseudo-metric and therefore $d_{\mathscr{S}}$ being the supremum of this family (in the lattice of pseudo-metrics over $\mathscr{X}$) is itself a pseudo-metric.

The first definition is important because it allows one to perform the OT computation in the low-dimensional space $\mathscr{Y}$, where it will be computationally less expansive.

Thus we have derived from $\mathscr{S}$ a pseudo-metric $d_{\mathscr{S}}$ on the space of probability measures $\mathscr{P}(\mathscr{X})$. We assume from now on that mappings in $\mathscr{S}$ are non-expansive. By point 4. above, we know that $d_{\mathscr{S}}$ is bounded above by $W(d_{\mathscr{X}})$. We call $d_{\mathscr{S}}$ the *Generalized Projected Wasserstein* metric (GPW) associated to $\mathscr{S}$. In good cases, it is both cheaper to compute and a good estimate.

3.3.3 SRW as an instance of GPW

In Paty and Cuturi (2019), the authors propose to estimate W_2 metrics by projecting the ambient Euclidean $\mathscr{X}$ into k-dimensional linear Euclidean subspaces. Specifically, their derived metric on $\mathscr{P}(X)$, written S_k, can be defined as (Paty & Cuturi, 2019, Th. 1, Eq. 4):

$$S_k^2(\mu,\nu) = \sup_{\Omega} W_2^2(d_{\mathscr{Y}})(\Omega_\#^{1/2}(\mu), \Omega_\#^{1/2}(\nu)) \tag{3.10}$$

where:

1. $d_{\mathscr{Y}}$ is the Euclidean metric on $\mathscr{Y}$
2. Ω contains all positive semi-definite matrices of trace k (and therefore admitting a well-defined square root) with associated semi-metric smaller than $d_{\mathscr{X}}$.

We recognise a particular case of our framework where the family of mappings is given by the linear mappings $\sqrt{\Omega} : \mathbb{R}^d = \mathscr{X} \to \mathscr{Y} = \mathbb{R}^k$ under the constraints above. In particular, all mappings used are linear. We confirm that by applying Algorithm 1 we essentially learn the Mahalanobis metric $\Omega = U * U^T$ defined in Section §2.2 and Weinberger and Saul (2009) with a stochastic gradient approach, using a 1-layer neural network architecture, with no activation function and no bias. The extra step of projecting back into the subspace of non-expansive neural networks can be achieved using with the spectral norm. In Figure 3.1 we can visualise the convergence curve of the authors approach & ours for 2 randomly sampled 20-D Gaussians with subspace of support $k = 5$.

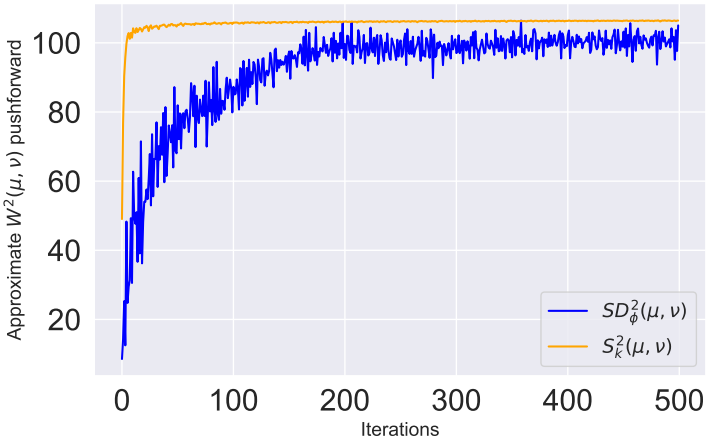


Figure 3.1: Estimation of $SD^2_\phi(\mu, \nu)$ using Algorithm 1 in blue over 500 iterations. Orange line shows exact computation of SRW distance $S^2_k(\mu, \nu)$

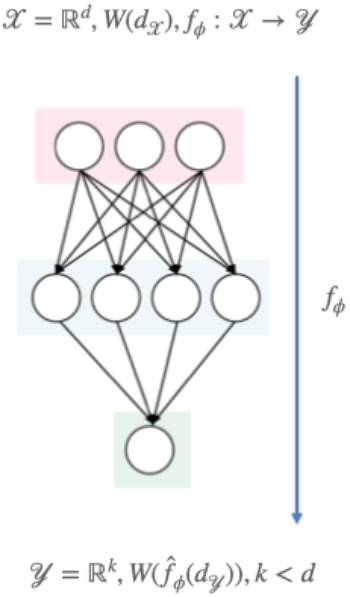


Figure 3.2: Illustration of the neural network mapping f_ϕ

The authors can complement the general properties of the approach with a specific explicit bound on the error and show that $S^2_k \leq W^2_2(d_{\mathcal{X}}) \leq (d/k)S^2_k$. In the general case, there is no upper bound available, and one has only the lower one.

As we will see in Section §3.4.2, we can devise a more efficient way to compute an approximation using a different family of projections, which we introduce next.

3.3.4 Non-linear embeddings for approximating Wasserstein distances

Using the same Euclidean metric spaces, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}^k$, we observe that our framework does not restrict us to use linear functions as mappings. One could use a family of mappings given by a neural network $(f_\phi : \mathcal{X} \to \mathcal{Y}; \phi \in S)$ where ϕ ranges over network weights. However, not any ϕ is correct. Indeed, by point 4) in the list of properties of $d_{\mathcal{S}}$, we need f_ϕs to be non-expansive. Ideally, we could pick S to be the set of all weights such that f_ϕ is non-expansive. We illustrate in Figure 3.2 the general constraints and framework for constructing f_ϕ.

There are two problems one needs to solve in order to reduce the idea to actual tractable computations. First, one needs an efficient gradient-based search to look for the weights ϕ which maximise $\sup_{\mathcal{S}} W(d_{\mathcal{Y}})(f_{\phi\#}(\mu), f_{\phi\#}(\nu))$ (see equation 3.8). Second, as the gradient update may take the current f_ϕ out of the non-expansive maps, one needs to project back efficiently in the space of 1-Lipschitz maps.

Both problems already have solutions which we have covered in Section §2.1.3, which we are going to re-use. For the first point, we will use the approximate Sinkhorn Divergence (SD) as presented in Genevay et al. (2017). Recent work by Feydy et al. (2018) shows that SD, which one can think of as a regularised version of W, is a sound choice as a loss function in machine learning, as outlined in Section §2.3. It can approximate W closely and without bias (see Genevay et al. (2017), Ramdas et al. (2017)), has better sample complexity (Genevay et al., 2019), as well as quadratic computation time. Most importantly, it is *fully differentiable*.

For the second problem, one can 'Lipshify' the linear layers of the network by dividing their (operator) norm after each update, a process known as spectral normalisation, introduced in the context of GANs by Miyato et al. (2018). We will use linear layers with Euclidean metrics, and this means we will need to estimate the spectral radius of each layer. The same could be done with linear layers using a mixture of L_1, L_2 and L_∞ metrics. In fact computing the $L_1 \to L_1$ operator norm for linear layers is an exact operation, as opposed to using the spectral norm for $L_2 \to L_2$ case in which we need to approximate using the power method. Note that the power method can only approximate the L_2 norm, and gradient ascent methods used in the maximization phase are stochastic, making our approximation susceptible to more variables. However, it is extremely efficient since it requires computation of optimal transport distances only in the low-dimensional space. We can see this as a trade-off between exactness and efficiency.

3.4 Experimental Evaluation

In this section, we consider the trade-offs and numerical differences between our proposed algorithm for GPW for estimating $W_2^2(d_{\mathscr{X}})$ and the algorithms proposed by Paty and Cuturi (2019)[Algorithm 1]. Their method requires the computation of an optimal transport distance at each iteration, whereas we take advantage of automatic differentiation and Sinkhorn Divergences. Both algorithms require a projection step, which in our case, introduces another variable to optimise for, namely the number of iterations during spectral normalisation. In their work, the initialisation of the algorithm is important for convergence, whereas we can initialise our neural network with random weights.

- First, we compare the Wasserstein estimates and the robustness and tradeoffs for a 10 dimensional Gausssian dataset.
- Secondly, we re-use our work to construct Algorithm 2 for training the discriminator of a GAN. Optimising the discriminator in our method is equivalent to learning an encoder f_ϕ.

Changing the loss function to unbiased Sinkhorn Divergences and performing spectral normalisation can be compared to weight clipping, as introduced by Arjovsky et al. (2017) to stabilise trainings of GANs. We train our GAN on a multi-modal 2D dataset with 8 Gaussian mixtures, which is known for being challenging and causing mode collapse for traditional generative methods.

3.4.1 Computational Details

We propose Algorithm 1 for stochastically estimating $d_{\mathscr{S}}$ between two measures with finite support where the class of mappings $\mathscr{S}$ is as defined above. As mentioned, a variant of this algorithm can further be used during the training of a discriminator as part of a generative network with an optimal transport objective, similar to Genevay et al. (2017). We devise Algorithm 2 and cover preliminary results using this approach in Section §3.4.3.

The Sinkhorn Divergence alternative for $d_{\mathscr{S}}$ now uses Sinkhorn divergences as a proxy for OT (compare with equation 3.8):

$$\begin{aligned} SD_{\phi,\varepsilon}(\mu,\nu) = W_\varepsilon(d_{\mathscr{Y}})(f_{\phi\#}(\mu), f_{\phi\#}(\nu)) - \frac{1}{2}W_\varepsilon(d_{\mathscr{Y}})(f_{\phi\#}(\mu), f_{\phi\#}(\mu)) \\ - \frac{1}{2}W_\varepsilon(d_{\mathscr{Y}})(f_{\phi\#}(\nu), f_{\phi\#}(\nu)) \end{aligned} \tag{3.11}$$

where W_ε is the well-known Sinkhorn regularized OT problem (Cuturi, 2013) presented in Section 2.1. The non-parameterized version of the divergence has been shown by Feydy et al. (2018) to be an unbiased estimator of $W(\mu,\nu)$ and converges to the true OT distance when $\varepsilon = 0$. Their paper also constructs an effective numerical scheme for computing the gradients of the Sinkhorn divergence on GPU, without having to back-propagate through the Sinkhorn iterations, by using *auto-differentiation* and the *detach* methods available in PyTorch (Paszke et al., 2019). Moreover, work by Schmitzer (2019) devised an ε-scaling scheme to trade-off between guaranteed convergence and speed. This gives us further control over how fast the algorithm is. It is important to note that the minimization computation happens in the low-dimensional space, differently from the approach in Paty and Cuturi (2019), which makes our algorithm scale better with dimension, as seen in Section §3.4.2.

Feydy et al. (2018) established that the gradient of equation 3.11 w.r.t to the input measures μ,ν is given by the dual optimal potentials. Since we are pushing the measures through a differentiable function f_ϕ, we can do the maximization step via a stochastic gradient ascent method such as *SGD* or *ADAM* (Kingma & Ba, 2014). Finally, after each iteration, we project back into the space of 1-Lipschitz functions f_ϕ. For domain-codomain $L_2 \longleftrightarrow L_2$ the Lipschitz constant of a fully connected layer is given by the spectral norm of the weights, which can be approximated in a few iterations of the power method. Since non-linear activation functions such as *ReLU* are 1-Lipschitz, in order to project back into the space of constraints we suggest

to normalize each layer's weights with the spectral norm, i.e. for layer i we have $\phi_i := \phi_i/||\phi_i||$. Previous work done by Neyshabur, Bhojanapalli, McAllester, and Srebro (2017) as well as Yoshida and Miyato (2017) and Miyato et al. (2018) showed that with smaller magnitude weights, the model can better generalize and improve the quality of generated samples when used on a discriminator in a GAN. We note that if we let f_ϕ to be a 1-Layer fully connected network with no activation, the optimization we perform is very similar with the optimization done by Paty and Cuturi (2019), as seen in Figure 3.1. The space of 1-Lipschitz functions we are optimizing over is larger and our method is stochastic, but we are able to recover very similar results at convergence. Moreover, our method applies to situations where the data lives in a non-linear manifold that an f_ϕ such as a neural network is able to model.

The focus of the next section is to compare different numerical properties of the Subspace Robust Wasserstein distances introduced in equation 3.10 with our Generalized Projected Wasserstein Distances.

Algorithm 1 Ground metric parameterization through ϕ

Input: Measures $\mu = \sum_i^n \delta_{x_i} a_i$ and $\nu = \sum_j^n \delta_{y_j} b_j$, $f_\phi : \mathbb{R}^d \to \mathbb{R}^k$ 2-Layer network with dimensions $(d, 20, k)$ and 1-Lipschitz, optimizer $ADAM$, power method iterations λ, $SD_{\phi,\varepsilon}$ unbiased Sinkhorn Divergence.

Output: f_ϕ, $SD_{\phi,\varepsilon}$

Initialize:

lr,ε,λ, $f_\phi \sim \mathcal{N}(0,10)$, $Objective \leftarrow SD_\varepsilon(blur = \varepsilon^2, p = 2, debias = True)$

for $t \to 1, \ldots, maxiter$ **do**

$L \leftarrow -SD_{\phi,\varepsilon}(f_{\phi\#}\mu, f_{\phi\#}\nu)$ (pushforward through f_ϕ and evaluate SD in lower space)

$grad_\phi \leftarrow$ **Autodiff**(L) (maximization step with autodiff)

$\phi \leftarrow \phi +$**ADAM**$(grad_\phi)$ (gradient step with SGD and scheduler)

$\phi \leftarrow Proj^\lambda_{1-Lip}(\phi)$ (projection into 1-Lipschitz space of functions)

end for

3.4.2 Empirical Analysis on Gaussian high-dimensional data

We consider similar experiments as presented in Forrow et al. (2019) and Paty and Cuturi (2019) and show the mean estimation of $SD^2_{\phi,k}(\mu,\nu)$ for different values of the latent dimension k, as well as robustness to noise. We also show how close the distance generated by the linear projector from Paty and Cuturi (2019) is to our distance and highlight the trade-off in terms of computation time with increasing number of dimensions.

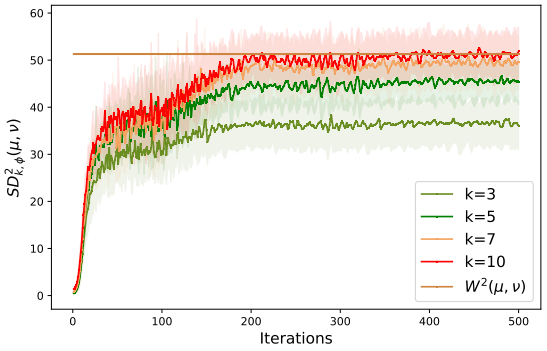


Figure 3.3: Mean estimation of $SD^2_\phi(\mu,\nu)$ for different values of the latent dimension k. Horizontal line is constant and shows the true $W^2(\mu,\nu)$. The shaded area shows the standard deviation over 20 runs.

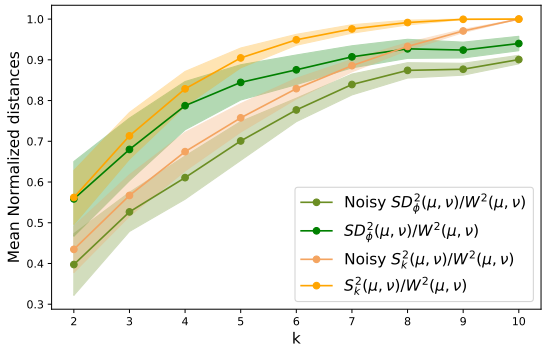


Figure 3.4: Mean normalized distances with and without noise for $SD^2_\phi(\mu,\nu)$ and $S^2_k(\mu,\nu)$ as a function of latent dimension k. The shaded area shows the standard deviation over 20 runs.

In order to illustrate our method, we construct two empirical distributions $\hat{\mu},\hat{\nu}$ by taking samples from two independent measures $\mu=\mathcal{N}(0,\Sigma_1)$ and $\nu=\mathcal{N}(0,\Sigma_2)$ that live in a 10 dimensional space. Similarly to Paty and Cuturi (2019) we construct the covariance matrices Σ_1,Σ_2 such that they are of rank 5, i.e. the support of the distributions is given by a 5 dimensional linear subspace. Throughout our experiments we fix f_ϕ to be a 2-layer neural network with a hidden layer of 16 units, activation function *ReLU* and output of dimension k. We initialize the weights from $\mathcal{N}(0,10)$ and use a standard *ADAM* optimizer with a decaying cyclic learning rate (Smith, 2017) bounded by $[0.1,1.0]$. Decreasing and increasing the learning rate via a scheduler allows us to not fall into local optima. The batch size for the algorithm is set to $n=500$, which is the same number of samples that make up the two measures.

Besides the neural network variables, we set the regularization strength small enough, to $\varepsilon=0.001$, and the scaling to ε-scaling $=0.95$ such that we can accurately estimate the true optimal transport distance, but not spend too much computational time during the Sinkhorn iterates.

10-D Gaussian Data OT estimation using $SD_{\phi,k}$

This leaves us with three variables of interest during the computation of $SD_{\phi,k}$, namely k,d,λ (latent dimension, input dimension, power method iterations). The power method iterations plays an important role during the projection step, as for a small number of iterations, there is a chance of breaking the constraint. At the same time, running the algorithm for too long is computationally expensive. In Figure 3.3 we used $\lambda=5$ power iterations and show the values of $SD^2_{k,\phi}$ after running Algorithm 1 for 500 iterations. We compare them to the true OT distance for various levels of k and observe that even with a small number of power iterations,

the estimation approaches the true value as k increases. Furthermore, we see that for $k = 5$ and $k = 7$ the algorithm converges after 200 steps.
Although we don't provide a lower bound for our algorithm, we expect the values to be similar to results from Paty and Cuturi (2019). We cover this in future work in Section §3.6.

Robustness

Using 20 power iterations, we show how the approximation behaves in the presence of noise as a function of the latent space k. We add Gaussian noise in the form of $\mathcal{N}(0, I)$ to $\hat{\mu}, \hat{\nu}$ and show in Figure 3.4 the comparison between not using noise and using noise for both SRW distances defined in equation 3.10 and GPW in equation 3.8. We observe that $SD^2_{\phi,k}$ behaves similarly to S^2_k in the presence of noise.

Computation time

In Figure. 8 of Paty and Cuturi (2019) they note that their method when using Sinkhorn iterates is quadratic in dimension because of the eigen-decomposition of the displacement matrix. Fundamentally different, we are always optimizing in the embedded space, making the computation of the Sinkhorn iterates *linear with dimension*. Note that there is the extra computation involved with pushing the measures through the neural network and backpropagating as well as the projection step that depends on the power iteration method. In order to run this experiment we set the power-method iterations fixed to $\lambda = 5$ and generate $\hat{\mu}, \hat{\nu}$ by changing dimension $d = [0, 2000]$ but leaving the rank of Σ_1, Σ_2 equal to 5. The latent space is fixed to $k = 5$.

In Figure 3.5 we plot the normalized distances using the two approaches as a function of dimension and see that the gap gets bigger with increasing dimensions, but it is stable. In Figure 3.6 we plot the log of the relative computation time, taking the $d = 10$ as a benchmark in both cases. We see that the time to compute SD^2_ϕ is linear in dimension and is significantly lower than its counterpart S^2_k as we increase the number of dimensions. This can be traced back to Algorithm. 1 and Algorithm. 2 of Paty and Cuturi (2019) where at each iteration step, the computation of OT distances in the data space is prohibitively expensive. Our method is purposefully designed to circumvent this.

3.4.3 Generative modelling with GPW

As covered in Section §2.3, using the unbiased version of Sinkhorn distances, namely Sinkhorn Divergences (Feydy et al., 2018) as a loss function for an adversarial generative model (GAN), can be a principled way to train such architectures. Since the discriminator's job is to push different samples further apart from each other, and similar ones, closer to each other, the min-max game from adversarial training using SD_ε as a loss, translates in finding a good ground metric on the data space. We can make use of Algorithm 1 & approaches from

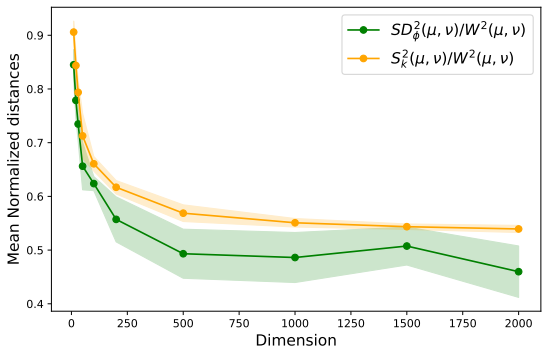


Figure 3.5: Comparison between normalized $SD^2_\phi(\mu,\nu)$ and normalized $S^2_k(\mu,\nu)$ as a function of dimension. The shaded area shows the standard deviation over 20 runs.

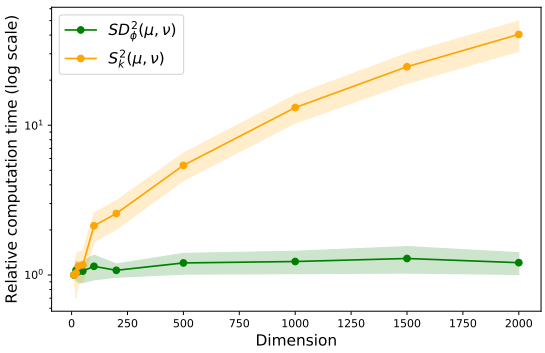
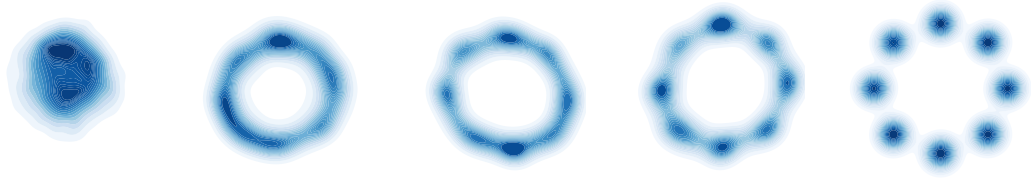


Figure 3.6: Mean relative computation time (log scale) comparison between the two distances. The shaded area shows shows the standard deviation over 20 runs.

Genevay et al. (2017) to construct a training algorithm, as seen in 2. As mentioned in Pop and Fulop (2018) and Srivastava et al. (2017) one of the main issues in training generative models is that of mode collapse, whereby the generator fails to capture the diversity of the training data distribution and gets stuck in a local minima.



(a) Epoch 1 **(b)** Epoch 5 **(c)** Epoch 10 **(d)** Epoch 20 **(e)** Real Data

Figure 3.7: Learning a 2D Mixture of Gaussians

One well known use-case for testing the stability of training for variants of generative adversarial networks is learning a multi-modal Gaussian distribution. We propose the standard Gaussian ring dataset with 8-mixtures of Gaussians in 2 dimensions and showcase in Figure 3.7 how using Algorithm 2, one can learn in 20 epochs, to generated a multi-modal distribution. The Generator and Discriminator are 2-layers dense networks, as described in Figure 3.8, with the Discriminator containing the spectral normalisation of layers.

$z \in \mathbb{R}^{10} \sim \mathcal{N}(0, I)$
Linear, 10×128, ReLU
Linear, 128×32, ReLU
Linear, 32×2

(a) Generator

Input $x \in \mathbb{R}^{100 \times 2}$
SNorm, Linear, 2×128, ReLU
SNorm, Linear, 128×32, ReLU
Linear, 128×10

(b) Discriminator f_ϕ

Figure 3.8: Linear Architectures for 2D Mixtures of Gaussian

3.5 Advances in Optimal Transport

One of the most used methods for approximating OT in high dimensions and reduce the complexity of the problem is to use Sliced Wasserstein Distances (SWD) (Kolouri et al., 2019). Recent work by Nguyen and Ho (2024); Nguyen et al. (2022) looks at improving the complexity of the solution by designing better projection mechanisms. In Nguyen and Ho (2024) the authors design a new distance variant, namely energy-based sliced wasserstein distances, which introduces a more structured approach to guide the projection process, by incorporating an energy-based model. They show favourable results for a range of applications from point-cloud gradient flows, color transfer, and deep point-cloud reconstruction.

Similarly, authors in Bonet, Courty, Septier, and Drumetz (2021) introduce the use of SWD for estimating gradient flows on euclidean spaces and in Bonet et al. (2022) they propose methods for estimating OT using spherical SWD that work on spherical data representations, solving tasks such as density estimation for geographical data points. A potential direction for future work is to compare the embeddings learned using our approach on spherical data structures with those obtained from other methods, and see if the data naturally clusters in the embedding space.

In the context of generative modelling, recent methods outlined in (Ho, Jain, & Abbeel, 2020; Rombach, Blattmann, Lorenz, Esser, & Ommer, 2022) rely on denoising diffusion probabilistic models and have shown great promise in the field of image synthesis and conditional generation. They can accurately learn a data distribution as well as provide a better, more accurate guiding mechanism for sampling.

In Rombach et al. (2022), authors propose latent diffusion models for image generation, a new class of models that applies diffusion in a latent space, significantly reducing the computational complexity, altogether maintaining high sample quality. A potential future direction would be to use the latent space learned by approximating OT distances as a base for applying diffusion. In Z. Li et al. (2023), authors propose using optimal transport to guide the reverse diffusion process, ensuring that each step in the sample generation is more efficient and better aligned with the target distribution, thus reducing the number of steps needed to transform noise into high-quality data.

3.6 Conclusions & Future Work

In this chapter we presented a new framework for approximating optimal transport distances using a wide family of embedding functions that are 1-Lipschitz. We showed how linear projectors can be considered as a special case of such functions and proceeded to define neural networks as another class of embeddings. We showed how we can use existing tools to build an efficient algorithm that is robust and constant in the dimension of the data. Furthermore, we showcased this approach by training a discriminator and generator on multi-modal synthetic data.

Future Work

Future work includes showing the approximation is valid for datasets where the support of distributions lies in a low-dimensional non-linear manifold, where we hypothesize that linear projects would fail. Extending the experiments, another avenue would be to evaluate how the convergence and robustness changes as we set the regularization strength higher and the ε-scaling smaller, to move more towards an MMD type of distance, rather than a Wasserstein one (see Feydy et al. (2018)). Trading off exactness to speed, can still provide good results for a discriminator training.

Other potential work includes experimenting with different operator norms such as L_1 or L_{inf} for the linear layers and the approximation of W_1. An extension of the projection step in 1 to convolutional layers would allow us to experiment with real datasets such as CIFAR-10 and learn a discriminator and naturally, a ground metric, in an adversarial way with $SD_{k,\phi}$ as a loss function. This can be used to show that the data naturally clusters in the embedding space.
On the theoretical side, future work includes proving that the lower-bound of our GPW estimates matches the lower bound, as presented in Paty and Cuturi (2019).

Algorithm 2 GAN Training using Sinkhorn Divergences

Input: Data distribution $p(D)$ a mixture of Gaussians,
Discriminator $f_\phi : \mathbb{R}^d \to \mathbb{R}^k$ with $n_d = 5, d = 2, k = 10$,
Generator $g_\theta : \mathbb{R}^k \to \mathbb{R}^d$, Optimizer $ADAM$, power method iterations λ,
$SD_{\phi,\varepsilon}$ unbiased Sinkhorn Divergence loss for discriminator, $SD_{\theta,\varepsilon}$ unbiased Sinkhorn Divergence loss for generator
Output: f_ϕ, g_θ
Initialise:
lr,ε,λ, $f_\phi \sim \mathcal{N}(0,d)$, $g_\phi \sim \mathcal{N}(0,k)$, $SD_\varepsilon(blur = \varepsilon^2, p = 2, debias = True)$, number of training iterations T as batch size $\times$ epochs.
for $t = 1, 2, \ldots, T$ **do**
Training Discriminator, freeze generator
$\mu \sim p(D)$ (sample real data batch)
$\nu \sim g_\theta(z)$, $z \sim \mathcal{N}(0,k)$ (generate noise data batch)
$L_1 \leftarrow -SD_{\phi,\varepsilon}(f_{\phi\#}\mu, f_{\phi\#}\nu)$ (pushforward through f_ϕ and compute discriminator loss)
$grad_\phi \leftarrow$ **Autodiff**(L_1) (maximization step with autodiff)
$\phi \leftarrow \phi +$**ADAM**$(grad_\phi)$ (gradient step with SGD and scheduler)
$\phi \leftarrow Proj^{\lambda}_{1-Lip}(\phi)$ (projection into 1-Lipschitz space of functions)

Training Generator, freeze discriminator
if $t \mod n_d == 0$ **then**
$\mu \sim p(D)$ (sample real data batch)
$\nu \sim g_\theta(z)$, $z \sim \mathcal{N}(0,k)$ (generate noise data batch)
$L_2 \leftarrow -SD_{\theta,\varepsilon}(f_{\phi\#}\mu, f_{\phi\#}\nu)$ (pushforward through f_ϕ and compute generator loss)
$grad_\theta \leftarrow$ **Autodiff**(L_2) (maximization step with autodiff)
$\theta \leftarrow \theta +$**ADAM**$(grad_\theta)$ (gradient step with SGD and scheduler)
end if
end for

Chapter 4

Ground Metric Learning

Metric Learning has been a longstanding problem in the field of machine learning, with multiple applications, most notably in image retrieval. Following the work of Rubner, Tomasi, and Guibas (2000) who first solved this task using the Earth Mover's Distance (Wasserstein-1) and noticed that perceptual similarity is preserved much better, we investigate a multiscale approach towards learning a distance metric on a high dimensional dataset using optimal transport distances. Due to the high cost of computing OT distances in high dimensions, we use PCA & VAE embeddings and extend the work of Cuturi and Avis (2014) for finding a ground metric. We provide an analysis into the benefits of using the pullback metric through the embedding space as an initialiser for the GML algorithm, as well as a comparison for the performance of image retrieval using the learnt pullback metrics in comparison to metrics such as LMNN.

4.1 Introduction

This chapter presents a multiscale approach towards learning a distance metric on a high dimensional dataset by learning the computationally cheaper pullback metric through a transformation of the original space. Specifically, we analyse learning the ground metric distance of optimal transport distances, or Wasserstein distances, by projecting on to the latent space obtained by linear transformations such as PCA or Autoencoders such as VAE (Kingma, Welling, et al., 2019).

We analyse the extension of the original Ground Metric Learning (GML) method presented in Cuturi and Avis (2014) by finding a good initialiser from learning a ground metric on pixel space in a lower dimensional space. The computational cost associated to finding optimal transport distances for distributions that live in high dimensions is prohibitively expensive, as the number of samples required to represent the space increases exponentially, one of the reasons being the curse of dimensionality (see Appendix A.3). As computing Wasserstein distances scales exponentially with sample size Dudley (1969), performing operations in a lower dimensional space will be computationally cheaper. We adapt the original GML algorithm by approximating Wasserstein distances using the Sinkhorn algorithm and its improvements, following the work in Altschuler et al. (2017); Cuturi (2013); Feydy et al. (2018). We analyse the results of the

algorithm on MNIST based on the accuracy of image retrieval using the distance learnt, by training and evaluating a KNN classifier. Since K-nearest neighbour is entirely based on the distance between features, this is the best way to measure whether distances learnt are performant.

In Section §4.2 we cover related work and provide relevant background on discrete optimal transport and the details of the method for learning ground metrics. In Section §4.3 we cover the details of the dataset used and how we can construct a more efficient task of learning a ground metric by finding a good initialiser using the pullback on a lower dimensional space. We present in Section §4.4 the results of applying the GML algorithm with different metric initialisers and the results obtained by doing KNN on binary and multiclass classification, as well as provide a comparison to using LMNN. Finally, we tackle the method limitations and conclusions in Section §4.6.

Contributions The key contributions of this chapter are:

- extending the original ground metric learning algorithm to use a more efficient metric initialiser
- comparison of the aforementioned extension against traditional metric learners such as LMNN

4.2 Related Work

As mentioned in Section §2.2, most of the research on learning metric distances falls into two categories, one that aims at learning the parameters of a metric, and kernel learning, a non-parametric approach that aims to learn a kernel matrix.

In this chapter, we will focus on the first scenario, following the work of Rubner et al. (2000), which defines metrics on the probability simplex as either bin-by-bin distance or cross-bin distances and is the first to use the Wasserstein distance to improve the task of image retrieval. The variants of f-divergences available, such as Jensen-Shannon or Hellinger, measure the similarity between two probability distribution by comparing d pairs (p_1^i, p_2^i) of points, whereas cross-bin distances compare all d^2 points and are better at capturing information between distributions with different supports, prior knowledge or statistical co-occurrences. Recently, Cuturi and Avis (2014) has shown that one can learn an optimal ground metric (GML) on the data space, and his work on introducing the Sinkhorn algorithm (Cuturi, 2013) to make the Wasserstein distance smoother, has paved the way for the community to find further optimal transport solvers, and lead to the introduction of Sinkhorn Divergences in (Feydy et al., 2018), which can interpolate between the true OT distance and MMD.

One of the most popular methods for supervised metric learning that we'll benchmark GML against is LMNN (Weinberger & Saul, 2009). It learns the Mahalanobis distance, using a linear transformation of the space such that instances of the same class are brought closer together. In fact, the Mahalanobis distance can be seen as the pullback of the euclidean distance through the learnt linear transformation.

4.2.1 Discrete Optimal Transport & Sinkhorn Divergences

Following the background on optimal transport, introduced in Section §2.1, we adapt the notations to closely match the ones encountered in Cuturi and Avis (2014); Cuturi and Doucet (2014); Genevay et al. (2016). We introduce discrete optimal transport on $\mathscr{X} = \mathbb{R}$ between the probability vectors $\mathbf{r}, \mathbf{c} \in \Sigma_d$ representing two normalised histograms of pixel intensities. In this case the distance or cost function is a pseudo-metric matrix taking values in $\mathscr{M} = \{d_{\mathscr{X}} \in \mathbb{R}^{d\times d} | 1 \leq i,j,k \leq d, m_{ij} < m_{ik} + m_{jk}, m_{ii} = 0\}$. For $m_{ij} > 0$, $d_{\mathscr{X}}$ is called a metric matrix and it belongs to $\mathscr{M}_+$.

Finding the Wasserstein distance between the two histograms, $W(\mathbf{r}, \mathbf{c})$ amounts to finding a coupling $\mathbf{X} \in \mathbb{R}_+^{d\times d}$ between the two histograms $\mathbf{r}, \mathbf{c}$, such that the Frobenius inner product $\langle d_{\mathscr{X}}, \mathbf{X} \rangle = tr(d_{\mathscr{X}}^T \mathbf{X})$ is minimized. Because we know that $\mathbf{X}$ is a coupling, we have to satisfy the linear constraints $\mathbf{X}\, 1_d = \mathbf{r}$ and $\mathbf{X}^T\, 1_d = \mathbf{c}$. Any such coupling solutions will belong in the set $U(\mathbf{r}, \mathbf{c}) = \{\mathbf{X} \,|\, \mathbf{X}\, 1_d = \mathbf{r}, \mathbf{X}^T\, 1_d = \mathbf{c}\}$ which is known as the convex polyhedron, containing the set of points with a finite number of extreme points. These extreme points are $\mathbf{X}^*$, the coupling solutions. Hence $\mathbb{W}(\mathbf{r}, \mathbf{c}) = \min\limits_{X \in U(\mathbf{r},\mathbf{c})} \langle d_{\mathscr{X}}, \mathbf{X} \rangle = \langle d_{\mathscr{X}}, \mathbf{X}^* \rangle$ is the optimal transport distance as seen in Villani (2008) and Cuturi and Avis (2014).

Furthermore, we can write the regularized version of discrete OT using the dual formulation for the OT distance (see Rubinstein-Kantorovich duality in Appendix A.1) using the dual variables α, β, s.t. $\alpha(X) + \beta(Y) \leq d_{\mathscr{X}}(X,Y)$:

$$\mathbb{W}_\gamma(\mathbf{r}, \mathbf{c}) = \max_{\alpha,\beta \in \mathscr{X}} \langle \alpha(X) \rangle_{\mathbf{r}} + \langle \beta(Y) \rangle_{\mathbf{c}} - \gamma \sum_{X,Y} e^{\frac{1}{\gamma}(\alpha(X) + \beta(Y) - d_{\mathscr{X}}(X,Y)} \tag{4.1}$$

The original solution for Equation 4.1 is the original Sinkhorn algorithm introduced in Cuturi (2013) and covered in Appendix A.1. However, the most complete approximation for Wasserstein distances is the Sinkhorn divergences introduced in (Feydy et al., 2018) & Section §2.1. For the discrete case, we can rewrite it as:

$$S_\gamma(\mathbf{r}, \mathbf{c}) = \mathbb{W}_\gamma(\mathbf{r}, \mathbf{c}) - \frac{1}{2}\mathbb{W}_\gamma(\mathbf{r}, \mathbf{r}) - \frac{1}{2}\mathbb{W}_\gamma(\mathbf{c}, \mathbf{c}) \tag{4.2}$$

In both cases, $\gamma = \frac{1}{\varepsilon}$ is the regularisation parameter that controls whether the approximation is exact, or tends towards more the MMD distance.

4.2.2 Ground Metric Learning

In his seminal paper, Cuturi and Avis (2014) uses Wasserstein distances to learn a metric $d_{\mathscr{X}}$ on the pixel space. Given a set of labelled histograms $\{\mathbf{a}_1, \mathbf{a}_2, ... \mathbf{a}_n\} \in \Sigma_d$, similarly to work done by Weinberger and Saul (2009), he introduces w_{ij} as coefficients measuring similarity between $\mathbf{a}_i, \mathbf{a}_j$, taking into account the similarities/dissimilarities within a dataset. The aim of the algorithm is to recover a 'ground' metric that is in agreement with these labels.

Further, we fix histograms $\mathbf{r}, \mathbf{c}$ and let the Wasserstein distance be seen as a function of $d_{\mathscr{X}}$:

$$\begin{aligned} G_{\mathbf{r},\mathbf{c}}(d_{\mathscr{X}}) &= \mathbb{W}(d_{\mathscr{X}})(\mathbf{r}, \mathbf{c}) \\ &= \min_{\mathbf{X} \in U(\mathbf{r},\mathbf{c})} \langle d_{\mathscr{X}}, \mathbf{X} \rangle \end{aligned} \tag{4.3}$$

The properties of this function as covered by Cuturi and Avis (2014) will allow us to construct a subgradient descent method to find an optimum ground metric $d_{\mathscr{X}}$.

- $G_{\mathbf{r},\mathbf{c}}(d_{\mathscr{X}})$ is piecewise linear.
- $\mathbf{X}^*$ belongs to $\partial G_{\mathbf{r},\mathbf{c}}(d_{\mathscr{X}})$, the subgradient of G at $d_{\mathscr{X}}$
- $G_{\mathbf{r},\mathbf{c}}(d_{\mathscr{X}})$ is also concave being the point-wise minimum of a set of affine functions (Boyd & Vandenberghe, 2004, §3.2.3)

Because the distances and similarity matrices are symmetrical, we can define the subsets of indices referring to similar and dissimilar histograms $\varepsilon_+ = \{(i,j) \in \mathscr{I} | w_{ij} > 0\}$ and $\varepsilon_- = \{(i,j) \in \mathscr{I} | w_{ij} < 0\}$, using only the upper diagonal matrix that represents the possible histogram comparisons, $\mathscr{I} = \{(i,j) | i < j\}$. Furthermore, we restrict the set of metric matrices to the closed set of stochastic matrices, $\mathscr{M}^1$.

The criteria developed below for learning $d_{\mathscr{X}}$ should have a greater optimal distance between dissimilar histograms and smaller optimal distance for similar histograms. Transporting between the same histograms should cost us as little as possible and we should weigh our distances by the supervision of our labels. One way of doing this is to choose k neighbours that are close in terms of G for each histogram. In what follows, N_{ik}^+ is the set of k neighbours of histogram i that have $w_{ij} > 0$.

$$\begin{aligned} C_k(d_{\mathscr{X}}) &= \sum_i^n \sum_{j \in N_{ik}^+} w_{ij} G_{ij}(d_{\mathscr{X}}) + \sum_{j \in N_{ik}^-} w_{ij} G_{ij}(d_{\mathscr{X}}) \\ &= Q_k^-(d_{\mathscr{X}}) + Q_k^+(d_{\mathscr{X}}) \end{aligned} \tag{4.4}$$

Minimizing the above criteria and taking advantage of the properties of G, leads to an algorithm that efficiently finds the two summation terms in equation 4.4 and results in the below gradient:

$$\nabla C_k(d_{\mathscr{X}}) = \sum_i^n \sum_{j \in N_{ik}^+} w_{ij} \mathbf{X}_{ij}^* + \sum_{j \in N_{ik}^-} w_{ij} \mathbf{X}_{ij}^* \tag{4.5}$$

A second algorithm then uses a subgradient descent method for h steps, together with a Taylor expansion of the positive side using the gradient ψ_+ from equation 4.5 to find a good candidate metric matrix $d^h_{\mathscr{X}}$. The resulting metric is given by:

$$d^{h+1}_{\mathscr{X}} \in \underset{d_{\mathscr{X}}}{argmin} Q^-_k(d_{\mathscr{X}}) + Q^+_k(d^h_{\mathscr{X}}) + \psi^T_+(d_{\mathscr{X}} - d^h_{\mathscr{X}}) \tag{4.6}$$

4.3 Methodology

In their paper, Cuturi and Avis (2014) use as an example features extracted from images of the Caltech-256 dataset to learn the metric between the points. In the preprocessing phase, the histograms are transformed using the GIST features (Oliva & Torralba, 2006), that represent 8 edge directions using a 4×4 grid image. Their histogram dimension is thus 128. Out of the 256 classes they select 30 images/class for training the model and form a binary classifier for 1000 random pairs of classes. In this case, there are 60 histograms split into two classes. In the next section we follow the same approach as Cuturi, but using instead Sinkhorn divergences as seen in equation 4.2.

The neighbourhood parameter is chosen to be small, $k = 3$, meaning we calculate optimal distances for 3 neighbours. The w values come from the two classes and are set to be $w = +/- \frac{1}{nk}$. To validate this approach, they calculate the Wasserstein distance using the learned metric and show it results in better classifications for a k-nearest-neighbour classifier compared to off-the shelf distances such as L2, Hellinger and other learned metrics such as ITML Davis et al. (2007) and LMNN Weinberger and Saul (2009).

In this section we apply the metric learning algorithm in the binary and multiclass scenarios using low-dimensional representations of the data, specifically PCA and VAE representations of the MNIST digit dataset using Sinkhorn Divergences. MNIST contains 1800 images each of dimension 64 (8x8) which are preprocessed by first taking the Hellinger[1] representation and transforming each digit into a probability vector. We can interpret the processed data-points as histograms of pixel intensity values. Details of the experiment settings are outlined in Table 4.1.

1. As described by Amari and Nagaoka (2007) in their seminal book on information geometry, the pullback of Hellinger representation through Euclidean is Hellinger distance

Table 4.1: Experiment settings for **MNIST**

Dataset	Method	Training epochs	Data size train/test	Output size
MNIST	PCA	1	449/1348	36
MNIST	VAE	100	449/1348	25

The aim is to construct a ground metric $d_{\mathscr{Y}}$ that accurately represents similarities within the the pixel space, given the Wasserstein distance can be calculated from labelled data for which we have similarity information. The original ground metric is a 64x64 dimension matrix that is used to calculate optimal transport distances between pairs of images. The optimal ground metric $\mathbf{X}^*$ will itself be a 64x64 matrix. The original algorithm assumes 600 iterations which involve at each step the computation of $\frac{n(n-1)}{2}$ Wasserstein distances. This quickly becomes expensive with the amount of training data, so we propose to tackle this issue in two ways.

- We learn the ground metric $\mathbf{X}^*$ by using two low-dimensional representations of the original dataset by applying PCA transformations and learning a VAE and computing the pullback metric to initialize the algorithm on the original space. We achieve comparable results at a lower computational cost.
- We achieve lower computational costs by using the Sinkhorn Divergences as seen in Section §4.2. As covered in Feydy et al. (2018) and Section §3.4.3 we choose a very small regularisation parameter ($\gamma \to 0$) as well as ε-scaling of 0.95, to stay close to the true Wasserstein distance.

A standard *scikit-learn* PCA implementation is used and the VAE is implemented and trained using PyTorch (Paszke et al., 2019). In general, when evaluating metric learning algorithms, the approach is to compute the accuracy of retrieval of similar classes, which can be done with a KNN classifier. In order to show the performance of the metric learned through GML, we compare various values of k nearest neighbours retrieval to LMNN.

4.4 Experimental Evaluation

Initially, we showcase our results for the binary classification problem, i.e. learning the metric for pairs of digits from MNIST and averaging the KNN performance on 45 different pairs. Because of the small number of training points used for each class (30) this is a rather inexpensive problem. For a 60 point training set we would need to compute 1800 distances at each step of the algorithm.

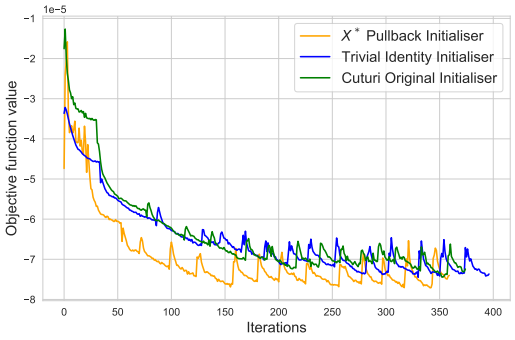


Figure 4.1: Objective for GML (Binary)

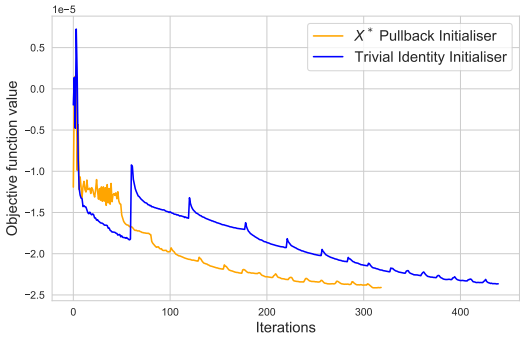


Figure 4.2: Objective for GML (Multiclass)

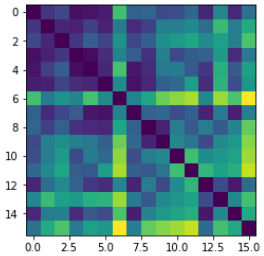


Figure 4.3: Ground metric Y^* on PCA space (16×16)

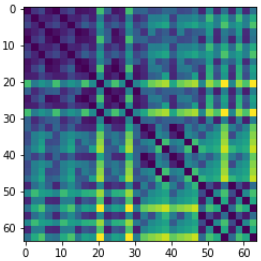


Figure 4.4: Binary Ground metric X^* (64×64)

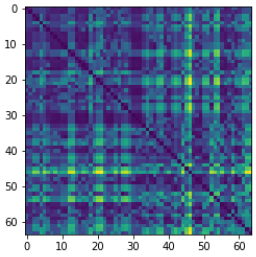


Figure 4.5: Multiclass Ground metric X^* (64×64)

The first step of the procedure is to solve the ground metric learning problem in the PCA or VAE transformed space and find a metric $\mathbf{Y}^* \in d_{\mathscr{Y}}$ within that space. By the Naturality concept covered in Section §3.3.1, we know that the pullback metric $\mathbf{X}^0 \in d_{\mathscr{X}}$ is the lowest scoring metric among the metrics that come from a lower dimensional space and can be used as an initialiser for GML.

Binary Classification

We perform the metric learning algorithm for digits 1 and 6 and visualise in Figure 4.1 how the algorithm learns the objective function in equation 4.6 based on three initialisation points, the trivial identity metric, the initialisation metric presented in Cuturi and Avis (2014) and our PCA pullback metric. We can see that the algorithm converges faster using the pullback as initialiser. In Figure 4.3 we showcase $\mathbf{Y}^*$, the lower-dimensional PCA space metric learnt as well as $\mathbf{X}^0$, the pullback metric in the binary case in Figure 4.4.

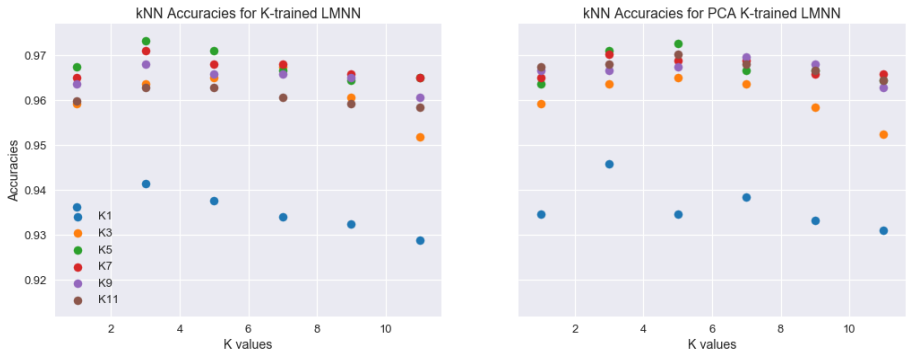


Figure 4.6: KNN accuracy trained for K neighbours LMNN on SIFT and PCA. **Left:** MNIST SIFT **Right:** MNIST PCA

Multiclass classification

Next, we perform the metric learning across the full range of MNIST and visualise in Figure 4.2 how the algorithm learns, based on two initialisers, namely the identity metric and the PCA pullback one. The learnt ground metric can be visualised in Figure 4.5. In order to evaluate the usefulness of the algorithm, we compare the accuracies of KNN on MNIST, between doing metric learning with LMNN and W-GML, on the original space and two lower dimensional spaces, i.e. PCA & VAE.

To start with, when training LMNN, the main hyper-parameter is number of neighbours retrieved at train time, something we refer to as the K-value. In Figure 4.6 we observe the difference between LMNN trained on the original SIFT (Scale-Invariant-Feature-Transform) representation of MNIST in comparison with the PCA representation, and notice that as expected, accuracies decrease with more neighbours retrieved at test time, but for training time, the optimal seems to be 5 to 7 neighbours. We don't notice major differences, however SIFT seems to produce slightly better results.

In Figure 4.8 we provide a comparison between KNN accuracies with LMNN and W-GML trained with k=3 neighbours for MNIST on the original SIFT space, on PCA space and VAE. We notice that GML and LMNN are comparable whilst we are using the SIFT space, whilst performance degrades when using a smaller learnt space such as VAEs.

In Figure 4.7 we have very similar results after initializing with the pullback-PCA metric during the training of the metric learner. For LMNN it takes considerable less time to train with the new initialisation of the pullback metric, as opposed to GML where it takes about half the time in comparison to using the identity metric. The training times for LMNN with different values of k using the identity matrix as an initialiser takes much longer than training k-LMNN on the PCA pullback representation (see table 4.2).

Table 4.2: Time (s) comparison for MNIST LMNN training initialized with **Identity** and **pullback PCA**

LMNN Initializer	K=1	K=3	K=5	K=7
Identity	64	179	269	363
Pullback PCA	0.05	0.11	0.17	0.23

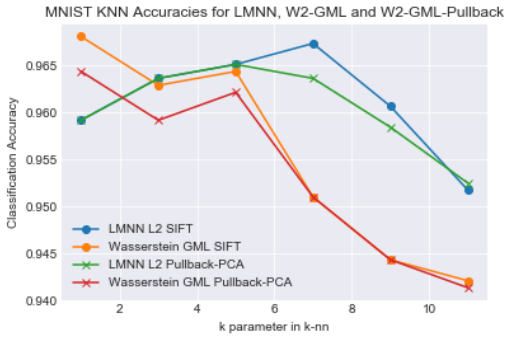


Figure 4.7: KNN accuracy trained for K neighbours on SIFT and Pullback-PCA

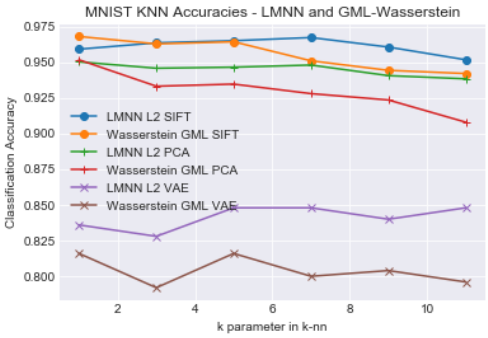


Figure 4.8: KNN accuracy trained for K neighbours on SIFT, PCA and VAE

4.5 Advances in Metric Learning & Image Retrieval

In recent years, traditional metric learning methods have been overtaken by approaches using deep neural networks, that can scale much better with the amount of data used. As opposed to learning a metric function or a kernel, these recent deep metric learning (DML) methods aim to capture the inherent relationships between data points by mapping them into a lower-dimensional space, where the distances between these neural network embeddings accurately reflect the semantic distances in the original space. The recent reviews by Ghojogh, Crowley, Karray, and Ghodsi (2023); Ghojogh, Ghodsi, Karray, and Crowley (2022) cover the main approaches in the literature, including different types of architectures such as Siamese networks and types of losses, from traditional cross-entropy, to contrastive losses, triplet losses and proxy-based methods such as Proxy-NCA. All of these methods enforce constraints in the embedding space, such that similar points are clustered together, and dissimilar ones are separated.

With diverse applications and across different domains, the most relevant ones to the work presented in this thesis are fine grained image recognition and image retrieval, few-shot and zero-shot learning or clustering. DML is a supervised method that requires labels to train a model. These labels are often costly to acquire and these methods are mostly used on specific tasks.

The more recent techniques of self-supervised learning are unsupervised learning methods that create more general representations using convolutional neural networks or vision transformers and can transfer well across domains. They rely on intrinsic properties of the data to create general-purpose representations. Early revolutionary methods such as SimCLR Chen, Kornblith, Norouzi, and Hinton (2020) use augmentations and contrastive losses to compare similar and dissimilar pairs of images and create general representations further used for downstream tasks.

For tasks such as fine-grained image recognition and image retrieval (Wei et al., 2021), where the intra-class variation is high, the performant current approaches involve the use of self-supervised models in combination with DML for specificity. The early work of El-Nouby, Neverova, Laptev, and Jégou (2021) proposed the use of a vision transformer backbone to extract general image descriptors and DML techniques on top of the representations to create superior results on category-level image retrieval, such as Stanford Online Product, In-Shop and CUB-200. However, more recent work hints at building image embeddings that are general enough to solve fine-grained challenges as well. In Ypsilantis et al. (2023) the authors build a cross-domain large dataset (8 domains, 349,000 classes) with the aim of creating a universal image embedding, capable of accurate image retrieval on fine-grained datasets as INaturalist or Cars196.

The early work of authors in Caron et al. (2020) focus on learning general visual representations with SSL by clustering different views of the same image. By re-framing the problem of clustering in the embedding space as an optimal transport problem from the data samples to the cluster centers, they can take into account the inherent metric of the embedding space and utilise the Sinkhorn-Knopp algorithm, thus resulting in more powerful visual representations. Finally, the work of DINO and DINOv2 (Caron et al., 2021; Oquab et al., 2023) unilaterally showed the power of self-supervised learning for constructing generalisable image embeddings powerful enough to be used in downstream tasks for image segmentation, depth estimation and image retrieval, including on fine-grained datasets (Flowers, Cars, Food).

The generalisability of SSL techniques used in conjuction with optimal transport approaches for better feature alignment is further validated by applications. In Izquierdo and Civera (2024) authors solve the task of Visual Place Recognition (VPR) by re-using a DINOV2 backbone as an image encoder and utilising optimal transport methods to further cluster local features into general image descriptors. Their one-stage method performs superior in comparison with baselines adding better geometric and temporal consistency.

4.6 Conclusion

In this chapter we presented an approach for learning the ground metric on the grid space of pixels and obtain more accurate optimal transport distances, in similar fashion to Cuturi and Avis (2014). We showcased the performance of learned metrics by evaluating the image retrieval task and compared against traditional such as LMNN on the MNIST dataset. The computational performance gained by using low-dimensional representations of the data such as PCA or VAE, did not directly translate into constant performance in terms of accuracy.

Challenges & Limitations

The main body of work presented in this chapter, can be traced back to finding a better initialiser for the GML algorithm in Cuturi and Avis (2014). The method presents some challenges & limitations that we'll briefly cover and use as motivation for the work presented in Chapter 3.

One of the key aspects of the approach presented is that it learns a ground metric on the pixel space, or the support space where real data is defined. In the context of machine learning applications in the real world, one usually wants to learn a parametrised distribution on the data. To this extent, the ground metric would then be defined between samples from the data space, seen as samples from a wider data distribution, not between the support space. In this situation, the ground metric we have shown in Figure 4.4 & Figure 4.5 would be impossible to compute. However, one can parametrise the ground metric and learn a continuous transformation instead of a fixed grid, and this is the main motivation behind Section §3.4.3. This is similar to the work done by Huang et al. (2016) on applying Wasserstein distances to documents, seen as probability distributions over word embeddings.

At the time of writing and when the experiments occurred, the literature and implementation on Sinkhorn divergences was still new, with no easy way to create a differentiable method to replace the projected gradient ascent in GML. For these reasons, although we use Sinkhorn divergences which are cheaper to compute, we did not adapt the algorithm and optimisers to use the differentiability component, instead relying on sub-gradient approximations which are not as exact. We however cover this in Chapter 3.

Finally, the projected gradient ascent algorithm in Cuturi and Avis (2014) was not guaranteed to output a metric matrix, in particular one that obeys the triangle inequality. For this to happen, the algorithm relies on the sub-process of metric-nearness outlined in Sra, Tropp, and Dhillon (2004). Because this was 20 years old and written in C, there were numerous technical challenges associated to integrating this component.

Chapter 5

Conclusions

5.1 Future Directions

This thesis attempts to simplify the problem of finding metrics on high dimensional data by combining tools from optimal transport, generative modelling & metric learning. At a conceptual level, we design algorithms that are efficient and robust,

We identify a few areas of interest for future directions to explore. In the context of approximating the OT distances using Sinkhorn Divergences and projections, as well as learning metrics, we propose:

1. Extending the work on generative models with Sinkhorn Divergences to higher resolution images such as CIFAR-10/CIFAR-100, CUB-200-2011 for fine-grained visual categorisation and CelebA for face recognition. Are the generative models expressive enough and does the learned metric through discriminator provide any insights into the performance of image retrieval?
2. The use of deeper architectures and convolutional neural networks to use during the generative training algorithm. How well do these convolutional networks preserve the OT approximation and how does the convergence of the algorithms depend on the dimensionality of the input?
3. Does the metric learnt through the Discriminator trained in the Sinkhorn Divergence make sense for image retrieval?
4. Investigating the relationship between the number of power iterations in the spectral normalisation and the regularisation values in the divergence, to better understand under what conditions the training can be stabilised, given a use case.
5. Investigate the use of deep architectures such as Resnets as embedding extractors and applying the metric learning algorithm on the last few layers. The assumption here is that classification models are already good representation learners. Learning a better embedding at a low cost can then have impact in downstream applications such as few-shot learning or domain adaptation.

6. At a more theoretical level, proving that the metric learning approach presented in Chapter §3 actually learns a metric in the true sense, not just an approximation, would be an avenue to explore further. We have the 'Lipschifying' with the spectral norm approach that brings the metric within reasonable bounds, otherwise it will explode and won't be a metric, but we we would need to ensure the three metric properties are satisfied. This is similar to the approach taken by Pitis, Chan, Jamali, and Ba (2020). Finally, proving the lower bounds for the approximation.

5.2 Concluding Remarks

In this thesis, we have presented a framework and methods for using optimal transport distances to address challenges encountered in metric learning on high-dimensional data. First, we create a framework for Generalised Projection Wasserstein Distances, which allows us to approximate the true OT distance, through a linear or non-linear 1-Lipschitz map. Secondly, we construct algorithms to compute GPW using 1-Lipschitz neural networks as projection maps and Sinkhorn Divergences, using stochastic optimisation techniques. We show that we can efficiently approximate OT distances with such algorithms on synthetic data. We validate the method scales linearly with the size of the original dimension space, as opposed to exponentially, compared to previous methods. Thirdly, we showcase the robustness of these algorithms against noisy input data and extend the process to a generative adversarial network training algorithm with Sinkhorn Divergences, that is able to prevent mode collapse during learning whilst also learning an adversarial cost function. Finally, we extend the approach of ground metric learning of optimal transport distances, for discrete measures, by using the cheaper pullback metric as an initialiser. We evaluate these methods using Variational Autoencoders and PCA projections to learn the pullback metric and evaluate the efficacy of the metric learnt in the setup of image retrieval, against traditional metric learning methods such as LMNN. A theme throughout the thesis was to make use of the Sinkhorn Divergences and their properties, such as differentiability and low sample complexity during the implementation and evaluation phase.

Appendix A

Chapter 2: Supplementary Information

A.1 From KP to DP

We sketch a proof that assumes a theorem by Rockafellar outlined in (Rockafellar, 1997, Sectionm §37, p.260) that we do not prove. For a coupling $\gamma \in \Gamma(\mu, \nu)$, we express this constraint by writing the inf as a sup.

Proof.

$$\sup_{\alpha,\beta} \int_{\mathscr{X}} \alpha d\mu + \int_{\mathscr{X}} \beta d\nu - \int (\alpha(x) + \beta(y)) d\pi = 0 \tag{A.1}$$

if $\gamma \in \Gamma(\mu, \nu)$ or ∞ if γ is not a coupling of μ, ν. We can add the above to (KP) and rewrite the following, by exchanging inf with sup, according to Rockafellar's theorem.

$$\begin{aligned} &\min_{\gamma} \int_{\mathscr{X}\times\mathscr{X}} d_{\mathscr{X}}(x,y) d\gamma + \sup_{\alpha,\beta} \int_{\mathscr{X}} \alpha d\mu + \int_{\mathscr{X}} \beta d\nu - \int (\alpha(x) + \beta(y)) d\gamma = \\ &\sup_{\alpha,\beta} \int_{\mathscr{X}} \alpha d\mu + \int_{\mathscr{X}} \beta d\nu + \inf_{\gamma} \int_{\mathscr{X}\times\mathscr{X}} d_{\mathscr{X}}(x,y) - (\alpha(x) + \beta(y)) d\gamma \end{aligned} \tag{A.2}$$

We can now use the same approach we used in the beginning and write the inf and the constraints on α and β separately.

$$\inf_{\gamma} \int_{\mathscr{X}\times\mathscr{X}} d_{\mathscr{X}}(x,y) - (\alpha(x) + \beta(y)) d\gamma = 0 \tag{A.3}$$

If $\alpha(x) + \beta(y) \leq d_{\mathscr{X}}(x,y)$, for all (x,y) in $\mathscr{X} \times \mathscr{X}$. Otherwise result is $-\infty$. This leads to finding the max of the sum w.r.t to their measures, as per the (DP). □

A.2 MLE minimizes KL divergence

Given the empirical distribution, $\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N}\delta_{x_n}$ and the target distribution p_θ we have that:

$$\begin{aligned}
\mathrm{KL}(\hat{\mathrm{p}}||\mathrm{p}_\theta) &= \sum_x \hat{p}(x) log \frac{\hat{p}(x)}{p_\theta(x)} \\
&= \sum_x \hat{p}(x) log \hat{p}(x) - \sum_x \hat{p}(x) log p_\theta(x) \\
&= -H(\hat{p}) - \frac{1}{N}\sum_x \sum_{n=1}^{N} \delta(x - x_n) log p_\theta(x) \\
&= -H(\hat{p}) - \frac{1}{N}\sum_{n=1}^{N} log p_\theta(x_n)
\end{aligned} \tag{A.4}$$

This results in:

$$\begin{aligned}
\underset{\theta}{argmin}\, \mathrm{KL}(\hat{\mathrm{p}}||\mathrm{p}_\theta) &= \underset{\theta}{argmin} - \frac{1}{N}\sum_{n=1}^{N} log p_\theta(x_n) \\
&= \underset{\theta}{argmax} \frac{1}{N}\sum_{n=1}^{N} log p_\theta(x_n) \qquad = MLE
\end{aligned} \tag{A.5}$$

A.3 Curse of dimensionality

The curse of dimensionality is illustrated best by the example of a D-dimensional unit hypercube in which you want to measure the density of points around a point x by using a smaller hypercube such that it contains a fraction f of the total of points. The expected edge of the hypercube will be $e_D(f) = f^{1/D}$. For 10 dimensions and $f = 0.01$ we calculate that the edge would be $0.001^{1/10} \sim 0.6$. What this means is that to measure the density of 1% of the points around x we need 60% of the cube. This means that our method is not local anymore. For a euclidean metric, it would not make sense to take into account such a big distance between points. Since you can relate clustering to gaussian mixture models and the usual euclidean distance, one can see why Euclidean distance is not good for high-dimensions.

A.4 KL Challenges

Intuitively, for a distribution $\mathbf{p}(X)$, the support $\mathbf{supp}(\mathbf{p})$ is given by the smallest subset of $\mathscr{X}$ where $\mathbf{p}$ is concentrated Santambrogio (2015)[1]. For example, the support of $\delta(X)$ is 0 or the support of $\mathbf{U}(a,b)$ is the interval (a,b). We note that KL is defined w.r.t to the same measure, so for a situation in which the space $\mathbf{Z}$ has smaller dimensionality than $\mathscr{X}$, the support for two distributions can be different, i.e. they are singular w.r.t one another. In this case the densities are not defined w.r.t to the same measure and the KL is not defined. KL is also asymmetric, so it is not a distance and can be infinite.

- Let $Z \sim U(0,1)$ on unit interval and $\mathbf{p}(X)$ distribution of $(0,Z) \in \mathbb{R}^2$ and $\mathbf{p}_\theta(X)$ of $(\theta,Z) \in \mathbb{R}^2$.
- We can see that for $\theta = 0$, $KL(\mathbf{p}\,||\,\mathbf{p}_\theta)$ is infinite, since support of $\mathbf{p}$ is not included in $\mathbf{p}_\theta$ in 0.
- Wasserstein distance is defined and is equal to $|\theta|$ in this scenario.

A.5 Relationship between KL and Mutual Information

Following Cover and Thomas (2012), we define the relationship between the mutual information between two marginals μ, ν and KL Divergence between the joint coupling γ and the product of the marginals.

$$\begin{aligned} KL(\gamma||\mu\nu) &= \sum_{x,y} \gamma log \frac{\gamma}{\nu\mu} = \sum_{x,y} \gamma log \frac{(\mu|\nu)\nu}{\mu\nu} \\ &= \sum_{x,y} \gamma log \frac{(\mu|\nu)}{\mu} = \sum_{x,y} \gamma log(\mu|\nu) - \sum_{x,y} \gamma log(\mu) \\ &= \sum_{x,y} \gamma log(\mu|\nu) - \sum_{x} \mu log(\mu) = H(\mu) - H(\mu|\nu) \\ &= H(\mu) - (H(\mu,\nu) - H(\nu)) = H(\mu) + H(\nu) - H(\gamma) = M(\mu,\nu) \end{aligned} \tag{A.6}$$

We conclude that $KL(\gamma||\mu\nu) = M(\mu,\nu)$, result proven using the conditional probability and conditional entropy relations.

---

1. $\mathbf{supp}(\mathbf{p})$ is the smallest closed subset $\mathbf{A}$ s.t. $\mathbf{p}(\mathscr{X} \setminus \mathbf{A}) = 0$

A.6 Maximum entropy principle and Entropy Regularisation

The principle of maximum entropy was derived as a generalization from physics from the well known Boltzmann distribution of gas velocities. Here, we are interested in treating the *MaxEnt* principle from an information theoretic point of view as illustrated in Cover & Thomas (Cover & Thomas, 2012, §12). The *MaxEnt* principle states that for a set of probability distributions following some constraints, the distribution that best represents those constraints is the one with maximum entropy. Moreover, the form its density follows is an exponential function of the form $p(x) = e^{\lambda_0 - 1 + \sum_K \lambda_k f(x)}$, with λ_i being calculated such that the below constraints are satisfied.

For a discrete case, we would have some constraints of the type:

- $p(x) \geq 0$
- $\sum_X p(x) = 1$
- $\sum_X p(x) f_k(x) = F_k$ (for $f_k(x) = x$ this mean $F_k = \mathbb{E}[x]$)

This is often used in forming priors, as advocated by Jaynes (1982), since in choosing the *MaxEnt* distribution, one chooses the least informative prior. For the constraint of probabilities summing to 1, one recovers the uniform distribution from the principle of *MaxEnt*. For a r.v. that has expectation 0 and variance σ^2 one recovers $N(0, \sigma^2)$.

For example, the multivariate Gaussian distribution is the distribution with maximum entropy, subject to having a specified mean and covariance. This is an important remark for why the regularisation in all optimal transport approaches we present is in fact in the form of the entropy of the coupling, since it can be seen exactly as a prior on the joint probability.

A.7 From Dual Formulation (DP) to Regularised Wasserstein using Sinkhorn algorithm

We note what the dual formulation for the primal problem is and then look at the regularized version by rewriting equation 2.10. The distance between two marginals, $\mu(X) = \sum_y \gamma_{x,y}$ and $\nu(Y) = \sum_x \gamma_{x,y}$ is given by $W(\mu, \nu) = \min_{\gamma \in \Gamma(\mu,\nu)} \sum_{x,y} \gamma_{x,y} D(x,y)$, which is equivalent to:

$$\max_{\alpha,\beta} \langle \alpha(x) \rangle_\mu + \langle \beta(y) \rangle_\nu \tag{A.7}$$

where $\alpha(x)$ and $\beta(y)$ are functions such that $\alpha(x) + \beta(y) \leq D(x,y)$.

The smoothed distance can be solved using Lagrange multipliers and the above dual formulation (Cuturi, 2013). Rewrite equation 2.10 as:

$$W_\varepsilon(\mu, \nu) = \min_{\gamma \in \Gamma(\mu,\nu)} \sum_{x,y} \gamma_{x,y} D(x,y) + \frac{1}{\varepsilon} \gamma_{x,y} log \gamma_{x,y} \tag{A.8}$$

Using Lagrangian multipliers and the dual formulation, we can write the Lagrangian:

$$L(\gamma,\alpha,\beta)=\sum_{x,y}\gamma_{x,y}D_{x,y}+\frac{1}{\varepsilon}\gamma_{x,y}log\gamma_{x,y}+\alpha(x)\Big(\sum_{y}\gamma_{x,y}-\mu(x)\Big)+\beta(y)\Big(\sum_{x}\gamma_{x,y}-\nu(y)\Big) \quad \text{(A.9)}$$

Solving for γ, use $\frac{\partial L_{x,y}}{\partial \gamma_{x,y}}=0$ and recover:

$$\frac{\partial L_{x,y}}{\partial \gamma_{x,y}}=D(x,y)+\frac{1}{\varepsilon}(1+log\gamma_{x,y})-\alpha(x)-\beta(y)=0 \quad \text{(A.10)}$$

which gives the solution:

$$\gamma_{x,y}=e^{\varepsilon(\alpha(x)+\beta(y)-D(x,y))-1} \quad \text{(A.11)}$$

The distance $D(x,y)$ can also be seen as a matrix transporting points from $\mathscr{X}\to\mathscr{Y}$, and we can write the element-wise exponential of that as: $K_{x,y}=e^{-\varepsilon D(x,y)-1}$. K is also referred to as the Gibbs Kernel with $\frac{1}{\varepsilon}$ being the 'temperature'. We can follow the same argument for rewriting the remaining exponentials as vectors $p(x)=e^{\varepsilon\alpha(x)}$ and $q(y)=e^{\varepsilon\beta(y)}$. Taking into account that $p\odot Kq=\mu$ and $q\odot K^Tp=\nu$ and by Sinkhorn's theorem, we can rewrite the solution as:

$$\gamma=\text{diag}(p)K\text{diag}(q) \quad \text{(A.12)}$$

Solving for p and q is the Sinkhorn matrix scaling algorithm and involves only matrix multiplication:

$$\begin{aligned} p&\to\mu/Kq\\ q&\to\nu/K^Tp \end{aligned} \quad \text{(A.13)}$$

We can also recover $\alpha^*(x)=-\frac{1}{\varepsilon}logp$, means that we have the optimal solutions for the regularized transport problem. One can see that with D set as the Euclidean distance, you recover the Gaussian kernel and are working in the usual Euclidean metric space. The conclusion is that an optimal coupling is a diagonal scaling of the Gibbs kernel. A very important thing to note, if we rewrite the minimisation objective, we are actually minimizing the KL divergence between γ and the Gibbs kernel. The Sinkhorn algorithm is nothing but iterative projections for KL as explained in Benamou, Carlier, Cuturi, Nenna, and Peyré (2015).

$$\min_{\gamma}KL(\gamma||K) \quad \text{(A.14)}$$

A.8 Variational Autoencoder

The following is adapted from (Blei et al., 2017, §2.2). For a parametrisable encoder with θ and decoder with ϕ we rewrite the log likelihood $logp(x)$ before doing maximum likelihood. Because of the hidden/latent variables z and their parameters, the problem is not easily solved (intractability - $p(x) = \int p(z)p(x|z)dz$).

In their paper Kingma and Welling (2013) refer to $q_\phi(z|x)$ as a probabilistic encoder, because this tells us how to generate the hidden vars z from x, $q_\phi(z|x)$ being the approximation to the true posterior which is given by Bayes Th. $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$. The decoder is denoted by $p_\theta(x|z)$ since this tells us how to reconstruct x given a hidden representation z.

We re-write the KL divergence between the approximate encoder and the real one and show it's equivalent to the below.

$$\begin{aligned} KL(q(z|x)||p(z|x)) &= \mathbb{E}_{q(z|x)}[log\frac{q(z|x)}{p(z|x)}] \geq 0 \\ &= \mathbb{E}_{q(z|x)}[log\frac{q(z|x)p(x)}{p(x|z)p(z)}] \geq 0 \\ &= \mathbb{E}_{q(z|x)}[log\frac{q(z|x)}{p(z)}] - \mathbb{E}_{q(z|x)}[logp(x|z)] + logp(x) \geq 0 \end{aligned} \tag{A.15}$$

At the last step the expectation doesn't make sense since p is independent of q. Rewriting this one gets:

$$\begin{aligned} logp(x) &= -\mathbb{E}_{q(z|x)}[log\frac{q(z|x)}{p(z)}] + \mathbb{E}_{q(z|x)}[logp(x|z)] + KL(q(z|x)||p(z|x)) \\ &= ELBO + KL(q(z|x)||p(z|x)) \end{aligned} \tag{A.16}$$

which is the conceptual result related to variational inference. Let's rewrite the result as in the original paper following the above intuition. From (A.15) and (A.16) we can rewrite the likelihood for an individual point as the sum between the KL divergence between the true posterior and approximate posterior and the variational lower bound (ELBO). Since the KL divergence is always positive we'll be effectively maximizing ELBO or minimizing the negative ELBO, using it as a loss function.

$$\begin{aligned} logp_\theta(x^i) &= KL(q_\phi(z|x^i)||p_\theta(z|x^i)) + \mathbb{E}_{q_\phi(z|x^i)}[p_\theta(x^i|z)] - \mathbb{E}_{q_\phi(z|x^i)}[log\frac{q_\phi(z|x^i)}{p_\theta(z)}] \\ &\geq \mathbb{E}_{q_\phi(z|x^i)}[p_\theta(x^i|z)] - \mathbb{E}_{q_\phi(z|x^i)}[log\frac{q_\phi(z|x^i)}{p_\theta(z)}] \\ &\geq \mathbb{E}_{q_\phi(z|x^i)}[p_\theta(x^i|z)] - KL(q_\phi(z|x^i)||p_\theta(z)) \end{aligned} \tag{A.17}$$

The reconstruction loss in the first term of (A.17) is the log-likelihood for our true decoder representation weighted by the distribution on the latent variables. The second term acts as a regulariser between our approximate representation (encoder distribution) and the prior on the latent space, which is the model assumption we make, taking $p_\theta(z) \sim \mathcal{N}(0, 1)$. Differentiating the loss function w.r.t the model parameters θ and the variational parameters ϕ drawn from certain distributions and using standard MCMC methods will exhibit high variance as seen in Blei et al. (2017).

Appendix B

Chapter 3: Supplementary Information

B.1 Subspace Robust Wasserstein Distance

In Paty and Cuturi (2019) they define the second order displacement matrix $d \times d$ which has the property that trace of the matrix multiplication equals the cost. Therefore, we can view OT as minimizing the trace of V_π (linearity for the integral)

$$V_\pi = \int (x-y)(x-y)^T d\pi(x,y) \tag{B.1}$$

Given an orthogonal projector on the space of possible low k-dimensional projections, they define PRW, the k-dimensional projection robust 2-Wasserstein distance:

$$P_k(\mu,\nu) = \sup_{E \in \mathscr{G}_k} \mathscr{W}(P_{E\#}\mu, P_{E\#}\nu) \tag{B.2}$$

Finally, the SRW, k-dimensional subspace robust 2-Wasserstein distance is:

$$S_k(\mu,\nu) = \inf_\pi \sup_{E \in \mathscr{G}_k} \Big[\int ||P_E(x-y)||^2 d\pi(x,y) \Big]^{1/2} \tag{B.3}$$

They then solve for optimal solutions of S_k using projected supergradient methods.

Bibliography

Abid, B. K., & Gower, R. (2018). Stochastic algorithms for entropy-regularized optimal transport problems. In *International conference on artificial intelligence and statistics* (pp. 1505–1512).

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive science*, *9*(1), 147–169.

Altschuler, J., Weed, J., & Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *arXiv preprint arXiv:1705.09634*.

Amari, S.-i., & Nagaoka, H. (2007). *Methods of information geometry* (Vol. 191). American Mathematical Soc.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein Gan. *arXiv preprint arXiv:1701.07875*.

Bassetti, F., Bodini, A., & Regazzini, E. (2006). On minimum kantorovich distance estimators. *Statistics & probability letters*, *76*(12), 1298–1302.

Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., & Peyré, G. (2015). Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, *37*(2), A1111–A1138.

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (Vol. 6). Athena Scientific Belmont, MA.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*(518), 859–877.

Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., & Pham, M.-T. (2022). Spherical sliced-wasserstein. *arXiv preprint arXiv:2206.08780*.

Bonet, C., Courty, N., Septier, F., & Drumetz, L. (2021). Efficient gradient flows in sliced-wasserstein space. *arXiv preprint arXiv:2110.10972*.

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., & Schoelkopf, B. (2017). From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Carlier, G., Duval, V., Peyré, G., & Schmitzer, B. (2017). Convergence of entropic schemes for optimal transport and gradient flows. *SIAM Journal on Mathematical Analysis*, *49*(2), 1385–1418.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, *33*, 9912–9924.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 9650–9660).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607).

Claici, S., Chien, E., & Solomon, J. (2018). Stochastic wasserstein barycenters. *arXiv preprint arXiv:1802.05757*.

Courty, N., Flamary, R., Habrard, A., & Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. In *Advances in neural information processing systems* (pp. 3730–3739).

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., & Shah, M. (2023). Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Csurka, G. (2017). Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (pp. 2292–2300).

Cuturi, M., & Avis, D. (2014). Ground metric learning. *Journal of Machine Learning Research*, *15*(1), 533–564.

Cuturi, M., & Doucet, A. (2014). Fast computation of wasserstein barycenters. In *International conference on machine learning* (pp. 685–693).

Dantzig, G. (1963). *Linear programming and extensions*. Princeton university press.

Dantzig, G. (2016). *Linear programming and extensions*. Princeton university press.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on machine learning* (pp. 209–216).

Deselaers, T., Keysers, D., & Ney, H. (2008). Features for image retrieval: an experimental comparison. *Information retrieval*, *11*, 77–107.

Dudley, R. M. (1969). The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, *40*(1), 40–50.

El-Nouby, A., Neverova, N., Laptev, I., & Jégou, H. (2021). Training vision transformers for image retrieval. *arXiv preprint arXiv:2102.05644*.

Fatras, K., Zine, Y., Flamary, R., Gribonval, R., & Courty, N. (2019). Learning with minibatch wasserstein: asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-I., Trouvé, A., & Peyré, G. (2018). Interpolating between optimal transport and mmd using sinkhorn divergences. *arXiv preprint arXiv:1810.08278*.

Forrow, A., Hütter, J.-C., Nitzan, M., Rigollet, P., Schiebinger, G., & Weed, J. (2019). Statistical optimal transport via factored couplings. In *The 22nd international conference on artificial intelligence and statistics* (pp. 2454–2465).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics New York.

Fulop, P., Manataki, A., Agachi, A., Capital, E., & Pop, P. (n.d.). Predicting survival after surgery for brain.

Fulop, P. M., & Danos, V. (2021). Efficient estimates of optimal transport via low-dimensional embeddings. *arXiv preprint arXiv:2111.04838*.

Genevay, A., Chizat, L., Bach, F., Cuturi, M., & Peyré, G. (2019). Sample complexity of sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics* (pp. 1574–1583).

Genevay, A., Cuturi, M., Peyré, G., & Bach, F. (2016). Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems* (pp. 3440–3448).

Genevay, A., Peyré, G., & Cuturi, M. (2017). Sinkhorn-autodiff: Tractable wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*.

Ghahramani, Z. (2004). Unsupervised learning. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, 72–112.

Ghojogh, B., Crowley, M., Karray, F., & Ghodsi, A. (2023). Deep metric learning. In *Elements of dimensionality reduction and manifold learning* (pp. 531–562). Springer.

Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2022). Spectral, probabilistic, and deep metric learning: Tutorial and survey. *arXiv preprint arXiv:2201.09267*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 ieee computer society conference on computer vision and pattern recognition (cvpr'06)* (Vol. 2, pp. 1735–1742).

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, *14*(8), 1771–1800.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, 6840–6851.

Huang, G., Guo, C., Kusner, M. J., Sun, Y., Sha, F., & Weinberger, K. Q. (2016). Supervised word mover's distance. In *Advances in neural information processing systems* (pp. 4862–4870).

Izquierdo, S., & Civera, J. (2024). Optimal transport aggregation for visual place recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 17658–17668).

Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, *70*(9), 939–952.

Johnstone, I. M., & Titterington, D. M. (2009). *Statistical challenges of high-dimensional data* (Vol. 367) (No. 1906). The Royal Society Publishing.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, *6*(4), 366–422.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Kolouri, S., Pope, P. E., Martin, C. E., & Rohde, G. K. (2018). Sliced-wasserstein autoencoder: An embarrassingly simple generative model. *arXiv preprint arXiv:1804.01947*.

Kolouri, S., Pope, P. E., Martin, C. E., & Rohde, G. K. (2019). Sliced wasserstein auto-encoders. In *International conference on learning representations.*

Kulis, B., et al. (2013). Metric learning: A survey. *Foundations and Trends® in Machine Learning*, *5*(4), 287–364.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., & Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, *30*.

Li, Z., Li, S., Wang, Z., Lei, N., Luo, Z., & Gu, D. X. (2023). Dpm-ot: a new diffusion probabilistic model based on optimal transport. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 22624–22633).

Lin, T., Fan, C., Ho, N., Cuturi, M., & Jordan, M. I. (2020). Projection robust wasserstein distance and riemannian optimization. *arXiv preprint arXiv:2006.07458*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Montavon, G., Müller, K.-R., & Cuturi, M. (2015). Wasserstein training of Boltzmann machines. *arXiv preprint arXiv:1507.01972*.

Musgrave, K., Belongie, S. J., & Lim, S.-N. (2020). Pytorch metric learning. *ArXiv*, *abs/2008.09164*.

Muzellec, B., & Cuturi, M. (2019). Subspace detours: Building transport plans that are optimal on subspace projections. In *Advances in neural information processing systems* (pp. 6917–6928).

Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017). Exploring generalization in deep learning. In *Advances in neural information processing systems* (pp. 5947–5956).

Nguyen, K., & Ho, N. (2024). Energy-based sliced wasserstein distance. *Advances in Neural Information Processing Systems*, *36*.

Nguyen, K., Ren, T., Nguyen, H., Rout, L., Nguyen, T., & Ho, N. (2022). Hierarchical sliced wasserstein distance. *arXiv preprint arXiv:2209.13570*.

Oliva, A., & Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, *155*, 23–36.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... others (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (pp. 8026–8037).

Patrini, G., Berg, R. v. d., Forre, P., Carioni, M., Bhargav, S., Welling, M., ... Nielsen, F. (2018). Sinkhorn autoencoders. *arXiv preprint arXiv:1810.01118*.

Paty, F.-P., & Cuturi, M. (2019). Subspace robust wasserstein distances. *arXiv preprint arXiv:1901.08949*.

Paty, F.-P., & Cuturi, M. (2020). Regularized optimal transport is ground cost adversarial. *arXiv preprint arXiv:2002.03967*.

Peyré, G., Cuturi, M., et al. (2017). *Computational optimal transport* (Tech. Rep.).

Pitis, S., Chan, H., Jamali, K., & Ba, J. (2020). An inductive bias for distances: Neural nets that respect the triangle inequality. *arXiv preprint arXiv:2002.05825*.

Pop, R., & Fulop, P. (2018). Deep ensemble bayesian active learning: Addressing the mode collapse issue in monte carlo dropout via ensembles. *arXiv preprint arXiv:1811.03897*.

Ramdas, A., Trillos, N. G., & Cuturi, M. (2017). On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, *19*(2), 47.

Rockafellar, R. T. (1997). *Convex analysis* (Vol. 11). Princeton university press.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10684–10695).

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, *40*(2), 99–121.

Salimans, T., Zhang, H., Radford, A., & Metaxas, D. (2018). Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*.

Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkäuser, NY*.

Schmitzer, B. (2016). Stabilized sparse scaling algorithms for entropy regularized transport problems. *arXiv preprint arXiv:1610.06519*.

Schmitzer, B. (2019). Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, *41*(3), A1443–A1481.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 815–823).

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., & Blondel, M. (2017). Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 ieee winter conference on applications of computer vision (wacv)* (pp. 464–472).

Smola, A. J., Gretton, A., & Borgwardt, K. (2006). Maximum mean discrepancy. In *13th international conference, iconip* (pp. 3–6).

Sra, S., Tropp, J., & Dhillon, I. (2004). Triangle fixing algorithms for the metric nearness problem. *Advances in Neural Information Processing Systems*, *17*.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., & Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, *30*.

Tonello, A. M., Letizia, N. A., Righini, D., & Marcuzzi, F. (2019). Machine learning tips and tricks for power line communications. *IEEE Access*, *7*, 82434-82452. doi: doi:10.1109/ACCESS.2019.2923321

Villani, C. (2008). *Optimal transport: old and new* (Vol. 338). Springer Science & Business Media.

Von Neumann, J., & Morgenstern, O. (2007). *Theory of games and economic behavior*. Princeton university press.

Weed, J., Bach, F., et al. (2019). Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *Bernoulli*, *25*(4A), 2620–2648.

Wei, X.-S., Song, Y.-Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., ... Belongie, S. (2021). Fine-grained image analysis with deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(12), 8927–8948.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, *10*(Feb), 207–244.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 1–40.

Xing, E. P., Jordan, M. I., Russell, S. J., & Ng, A. Y. (2003). Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems* (pp. 521–528).

Yoshida, Y., & Miyato, T. (2017). Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*.

Ypsilantis, N.-A., Chen, K., Cao, B., Lipovskỳ, M., Dogan-Schönberger, P., Makosa, G., ... Araujo, A. (2023). Towards universal image embeddings: A large-scale dataset and challenge for generic image representations. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 11290–11301).

Zhai, A., & Wu, H.-Y. (2018). Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*.