Computer Speech and Language 17 (2003) 153–172

# Modelling the uncertainty in recovering articulation from acoustics

Korin Richmond *, Simon King, Paul Taylor

Centre for Speech Technology Research, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, UK

Abstract

This paper presents an experimental comparison of the performance of the multilayer perceptron (MLP) with that of the mixture density network (MDN) for an acoustic-to-articulatory mapping task. A corpus of acoustic-articulatory data recorded by electromagnetic articulography (EMA) for a single speaker was used as training and test data for this purpose. In theory, the MDN is able to provide a richer, more flexible description of the target variables in response to a given input vector than the least-squares trained MLP. Our results show that the mean likelihoods of the target articulatory parameters for an unseen test set were indeed consistently higher with the MDN than with the MLP. The increase ranged from approximately 3% to 22%, depending on the articulatory channel in question. On the basis of these results, we argue that using a more flexible description of the target domain, such as that offered by the MDN, can prove beneficial when modelling the acoustic-to-articulatory mapping.

## 1. Introduction

### 1.1. Background

A successful method for inferring articulation from the acoustic speech signal would find many potential applications: low bit-rate speech coding, helping individuals with speech or hearing impairment by providing visual feedback during speech training, and the possibility ofi mproved

* Corresponding author. Tel.: +44-131-651-1769; fax: +44-131-650-4587.
E-mail address: korin@cstr.ed.ac.uk (K. Richmond).

automatic speech recognition. Therefore, it is unsurprising that researchers have been investigating the acoustic-to-articulatory mapping for several decades.

Much of this work has pursued an analytical approach, whereby an acoustic signal is subjected to mathematical analysis to yield the area function of a tube model that might have generated it. Many early attempts in particular took this approach. For example, Wakita (1979) attempted to infer the area functions for a model vocal tract for vowel sounds. Unfortunately, analytical methods have struggled with some fundamental difficulties. Certain types of speech sound have proved more challenging than others, such as where coupling of the nasal and oral cavities occurs. Perhaps most importantly, there is no independent means of assessing the performance of the inversion mapping. Moreover, certain inferred vocal tract configurations may not even be physiologically possible, let alone common in the speech of human speakers.

Researchers have also turned to articulatory synthesis models for help in studying and developing an inversion mapping. These may be used as part of a "mimic" algorithm, where the model parameters are iteratively adjusted to minimise cost functions based on the model output and the target acoustic signal (e.g., Shirai & Kobayashi, 1986). Synthesis models have also been used to generate a database of acoustic-articulatory vector pairs by sampling from the domain of the articulatory control parameters. Such a database can either be used directly for performing the inversion mapping (e.g., Atal, Chang, Mathews, & Tukey, 1978), or as training data for other empirical learning models. As an example of the latter, Rahim, Goodyear, Kleijn, Schroeter, and Sondhi (1993) used Mermelstein's articulatory model (Mermelstein, 1973) to generate data for training various MLPs.

Unfortunately, as with analytical methods, the use of articulatory synthesis models still leaves us with fundamental problems in terms of satisfactory assessment with respect to real human speech. It is also conceivable that limitations in either the accuracy or scope of the model itself could manifest themselves in a generated data set.

Human articulography data avoids artifacts resulting from limitations and inaccuracies in the generating model that might afflict synthetic data. What is more, we are not left with the same difficulties in assessing how an inversion method is really performing; we can evaluate the performance of an inversion algorithm by comparison with how the speaker actually articulated an utterance. As such, human articulographic data is a very useful resource for studying the inversion mapping.

Thanks to technologies such as X-ray microbeam (XRMB) cinematography and electromagnetic articulography (EMA), measured human articulatory data has become increasingly accessible. However, despite the potential advantages, only a relatively small number of studies where empirical learning models have been applied to measured articulatory data have been previously reported. These include extended Kalman filtering (Dusan, 2000), self-organising HMMs (Roweis, 1999), codebook methods (Hogden et al., 1996), and the MLP (Papcun et al., 1992; Zachs & Thomas, 1994). The studies that have been done have focused almost entirely on a restricted set of speech sounds. Therefore, attempting inversion for all speech sounds and for continuous speech, as described in the current paper, is a necessary research step in itself.

### *1.2. Approach to inversion*

The MLP is well known as a universal function approximator and has advantages in terms of efficiency compared with many models. For example, Rahim, Kleijn, Schroeter, and Goodyear

(1991) reported their MLP inversion system used only 4% of the memory required by the codebook they used for comparison, and was able to perform the mapping 20 times faster. In view of the promise shown by the MLP in the past for performing the inversion mapping, we considered it a worthwhile candidate for evaluation on a larger database of more realistic articulatory-acoustic data. In Section 3, we present our experience of using the MLP for performing the inversion mapping.

From this point we go on to address the implications of evidence that indicates that multiple articulatory configurations are able to produce the same acoustic signal. Lindblom, Lubker, and Gay (1979), for example, found that speakers whose jaw had been fixed in position by a bite block were nevertheless able to produce vowels with formants well within the range of variation observed under normal conditions. Another compelling example was provided by Roweis (1999). Using roughly 20 min of speech from each of 57 speakers recorded at the X-ray microbeam facility at Wisconsin University (Westbury, 1994), Roweis compiled a large data set of acoustic-articulatory vector pairs. He then took a reference point in acoustic space and showed how the articulatory points paired with the nearest thousand points in acoustic space could be spread widely throughout the articulatory domain, and even feature multimodal distributions.

If humans do use a range of different articulatory configurations to produce the same acoustic signal, then the inversion mapping is a classic example of what is termed an *ill-posed* problem, as the solution is potentially non-unique. In general, it raises the question of how an inversion algorithm should decide between all the possible articulatory configurations that might be associated with a given acoustic feature vector. In particular, we are led to question how the MLP deals with ill-posed problems.

It is well known that the outputs of an MLP trained under the sum-of-squares error function approximate the average of the target data points conditioned on the input vector. This is equivalent to performing unimodal regression, with a residual global variance over the whole training data set (Bishop, 1995). While this might be an appropriate solution for many situations, it can be problematic in the case of ill-posed mappings. Where multiple branches of the solution exist, the average of several correct target values is not legitimate. In addition, there is no indication (dependent on the input) of how the potential target values are distributed about the mean.

The mixture density network (MDN), introduced by Bishop (1994), represents a method for modelling arbitrary probability density functions over the target domain, conditioned on the input vector. With respect to the inversion mapping problem, the MDN can in principle provide a full and flexible description of how likely all possible articulatory configurations are given an acoustic input.

If the conditional distributions of target points in articulatory space conform to a unimodal Gaussian with a variance that does not depend in a systematic way on the acoustic input, then the least-squares trained MLP should provide a sufficient description. However, if the distributions of articulatory points have variance related to the acoustic input, or take a more complex form than a unimodal Gaussian, then the MDN should demonstrate an advantage over the MLP. It is the aim of the current paper to test this and to compare directly the MLP with the MDN on an acoustic-to-articulatory mapping task.

In this respect, the motivation for the approach taken in this paper differs from the general aim apparent in previous work. It is the intention here to explore the potential for explicitly modelling the uncertainty, or variance, around estimated articulator positions. This approach is in part

motivated by the view that, if inferred articulation is to provide useful application, we need to know how much confidence to ascribe to the accuracy of the inferred articulatory parameters at each point in time.

This paper first introduces the measured articulatory data that has been used in the experiments described, and explains how it was processed for use as training data for the neural networks. Next, we describe our implementation of an MLP for performing the inversion task and evaluate its performance compared with other systems reported in the literature. We then describe how MDNs have been applied to the same inversion problem. Finally, the characteristics of an MLP and an MDN for performing the acoustic-to-articulatory mapping on this data set are compared.

## 2. The inversion task

### 2.1. MOCHA

The multichannel articulatory (MOCHA) database has recently been recorded in the purpose built studio at the Edinburgh Speech Production Facility at Queen Margaret University College (Wrench & Hardcastle, 2000).

During speech, four data streams were recorded concurrently straight to computer: the acoustic waveform (16 kHz sample rate, with 16 bit precision) together with laryngograph, electropalatograph and electromagnetic articulograph data. The electromagnetic articulograph sampled the movement of receiver coils attached to the articulators in the midsagittal plane at 500 Hz. Coils were affixed to the top lip, bottom lip, bottom incisor, tongue tip, tongue body, tongue dorsum and velum. With each of these sensors providing $x$- and $y$-coordinates in the midsagittal plane, 14 channels of salient articulatory information were recorded in total. Additional coils were attached to the bridge of the nose and the upper incisor. However, the signals from these coils, which should have minimal movement relative to each other, were only used to provide reference points for an algorithm which processed the other EMA channels to correct for head movement.

The speakers were recorded reading a set of 460 British TIMIT sentences. These short sentences were designed to provide phonetically diverse material. They were chosen in an attempt to capture with good coverage the connected speech processes in English and thus to maximise the usefulness of the MOCHA database for speech technology and speech science research purposes.

The final release of the database will feature up to 40 speakers with a variety of regional accents. At the time of conducting the experiments described here, two speakers had been made available: one male with a Northern English accent and one female speaker with a Southern English accent. For this paper, the acoustic waveform and EMA data recorded for the second speaker (`fsew0`) were used.

### 2.2. Data processing

In order to render the raw articulatory and acoustic data into a format suitable for use with neural networks, several processing steps were carried out. First, filterbank analysis was per-

formed on the acoustic signal, using a Hamming window of 20 ms with a shift of 10 ms. For each time frame, the acoustic vector consisted of 20 melscale filterbank coefficients. These were normalised across all 460 utterances to lie within the range $[0.0, 1.0]$.

The articulatory feature vector comprised the $x$- and $y$-coordinates of the seven EMA sensor coils. In order to obtain an articulatory feature vector to pair with each acoustic vector, the EMA traces were downsampled to match the 10 ms shift rate of the acoustic feature vectors. Prior to downsampling, the signal was first lowpass filtered with a zero-phase FIR filter to lessen the effect of noise resulting from measurement error in the EMA machine. The articulatory feature vectors were normalised to lie in the range $[0.1, 0.9]$. This range was used because the logistic activation function of the output units in the MLP has unrealisable asymptotic limits of 0.0 and 1.0. All normalisation was performed as standard, using mean and standard deviation, then shifting and scaling for the data to lie within the desired range. Further details of the normalisation process can be found in Richmond (2002).

The files for each utterance of speaker `fsewO` contain an average of approximately 1.3 s of silence in total before and after the utterance. This compares with an average utterance length of 2.7 s. During silent stretches, the mouth can potentially take any configuration. This could pose a serious problem to network training, because given an acoustic feature vector representing silence, the network would be attempting to map to a large range of possible articulatory configurations. Therefore, care was taken to omit all data containing silence.

From the 460 utterances contained in the database of speaker `fsewO`, 368 files were included in the training set, 46 files in a validation set, while 46 files were put aside for the test set. The full training set contained 92,557 pairs of acoustic and articulatory feature vectors.

## 3. MLP inversion mapping

The feedforward MLP used to perform the inversion mapping is shown in Fig. 1. As indicated, this network featured 14 output units, one for each of the $x$- and $y$-coordinates of the seven EMA coil positions. The input layer of the network contained 400 units, which provided a context window of 20 frames of 20 filterbank coefficients each. The use of a context window in the acoustic input domain was similar to the approach of Papcun et al. (1992), and empirical evaluation has indeed found it to be beneficial (Richmond, 2002).

We used the Skeletonization algorithm (Mozer & Smolensky, 1989) to help identify a suitable number of hidden units. This pruning algorithm works by taking a fully trained MLP and identifying the node whose removal would result in the lowest increase in error on the training set. This node is then removed, and the network is further trained to compensate. The two steps of removing the least salient node and then retraining are repeated iteratively until removing a further node results in a deficit in network performance that it is not possible to recoup.

In this case, the initial MLP contained 50 hidden units. The best performance was observed in the network with only 38 units remaining; after this point, the network became progressively worse with the removal of each unit. This number of units was corroborated as approximately suitable by a separate experiment, in which a set of MLPs were trained and evaluated with a range of hidden layer and input layer sizes (Richmond, 2002).

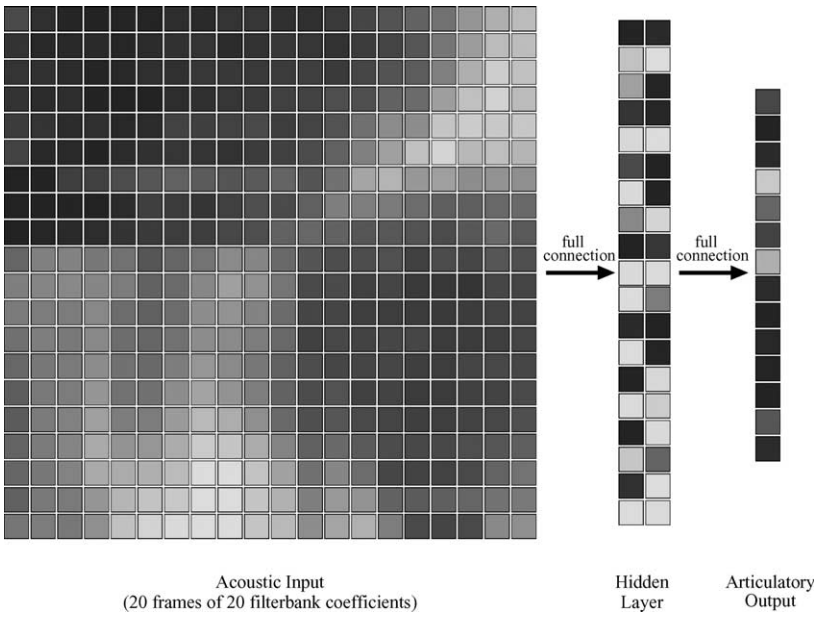


Fig. 1. Feedforward MLP for performing the inversion mapping. This figure shows the presentation of an acoustic input vector, made up of 20 time frames of 20 filterbank coefficients. Each frame of filterbank coefficients was computed with a shift of 10 ms, therefore the total context window of acoustic input applied to the network is approximately 200 ms. The hidden and output units used a logistic activation function.

As is common practice, the network shown in Fig. 1 was first initialised by randomising the weights to lie in the range $[-1.0, 1.0]$, and trained until the error calculated on the separate validation data set ceased to reduce. The scaled conjugate gradient (SCG) (Møller, 1993) optimisation algorithm was used to train the MLP. First order optimisation algorithms, such as standard backpropagation gradient descent, only make use of the first derivatives of the error function. Conjugate gradient optimisation algorithms on the other hand make use of the second derivatives of the error function. SCG has been shown to be considerably faster than standard backpropagation and other conjugate gradient methods (Møller, 1993). The SCG optimisation algorithm also has the convenience of not requiring the user to decide values for critical parameters, as all the algorithm's parameters are adaptive and the user only has to supply non-crucial initial values. This contrasts with simple gradient descent optimisation, for example, which is sensitive to the user supplied learning parameter $\eta$. When $\eta$ is small, the network may be slow to train and may become more easily stuck in local minima. Conversely, when $\eta$ is large, training can become erratic.

### 3.1. Results

Two measures that have been used in the past to compare trajectories of articulatory movements are root mean square (RMS) error and correlation. RMS error is an indication of the overall 'distance' between two trajectories. The correlation score is an indication of similarity of shape and synchrony of two trajectories.

Table 1
MLP performance when recovering articulation from acoustics for the unseen test set, given as RMS error and correlation for each articulatory channel

| Articulator | RMS error | RMS error (mm) | Correlation |
|---|---|---|---|
| Upper lip $x$ | 0.18 | 0.99 | 0.58 |
| Upper lip $y$ | 0.14 | 1.16 | 0.72 |
| Lower lip $x$ | 0.16 | 1.21 | 0.61 |
| Lower lip $y$ | 0.14 | 2.73 | 0.75 |
| Lower incisor $x$ | 0.16 | 0.89 | 0.56 |
| Lower incisor $y$ | 0.12 | 1.19 | 0.80 |
| Tongue tip $x$ | 0.13 | 2.43 | 0.79 |
| Tongue tip $y$ | 0.12 | 2.56 | 0.84 |
| Tongue body $x$ | 0.12 | 2.19 | 0.81 |
| Tongue body $y$ | 0.11 | 2.14 | 0.83 |
| Tongue dorsum $x$ | 0.13 | 2.04 | 0.79 |
| Tongue dorsum $y$ | 0.14 | 2.31 | 0.71 |
| Velum $x$ | 0.13 | 0.42 | 0.79 |
| Velum $y$ | 0.13 | 0.41 | 0.77 |

The average of the RMS error values given is 1.62 mm.

Table 1 presents the results for the MLP when estimating articulation for the unseen test set containing 46 utterances. The first column of this table shows RMS error, calculated separately for each of the 14 articulator channels as

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - t_i)^2}, \tag{1}$$

where $N$ is the number of input–output vector pairs, or patterns, in the test set, $o_i$ is the estimated value for the articulator channel output by the network, and $t_i$ is the measured value. The second column gives the same RMS errors, but scaled back to the original domain of EMA measurement in millimetres. Finally, the third column gives a measure of correlation between the actual articulatory trajectories and those estimated by the MLP. The measure is calculated by dividing their covariance by the square root of the product of their variances

$$r = \frac{\sum_i (o_i - \bar{o})(t_i - \bar{t})}{\sqrt{\sum_i (o_i - \bar{o})^2 \cdot \sum_i (t_i - \bar{t})^2}}, \tag{2}$$

where $\bar{o}$ and $\bar{t}$ are the mean channel value for the network output and actual articulator position respectively.

The results given in Table 1 compare favourably with those of other inversion mapping methods previously reported in the literature: Hogden et al. (1996) reported an error around 2 mm for the tongue points; Dusan (2000) reported average RMS error of around 2 mm; Okadome, Suzuki, and Honda (2000) reported RMS error between estimated and actual articulatory trajectories of about 1.8 mm on average. In addition, Richmond (2002) reported on using global linear mappings for exactly the same task with a range of different context window sizes. The

linear mapping with 20 frames within the context window performed with an error of 1.89 mm on average.

RMS error and correlation scores provide a means to compare objectively and quantitatively the performance of two different networks, or indeed different inversion techniques described previously in the literature. Meanwhile, visual comparison of MLP output with the measured articulatory trajectories undoubtedly gives a very useful qualitative impression of how the MLP is performing the inversion mapping, and can reveal clues as to the nature of error in the MLP output.

Fig. 2 demonstrates the output of the MLP for the test set utterance "The speech symposium might begin on Monday". The figure provides a direct comparison of the trajectories estimated by the MLP with those measured by EMA and normalised. Silence at the beginning and end of the files has been omitted. As this example shows, the MLP is typically capable of estimating the trajectories of some articulators with a good level of accuracy at some times, but less so at other times. For example, at around 1.8 s, the MLP estimates for the $x$-coordinates of the three tongue points are more accurate than for the $y$-coordinates of the same points. During the rest of the utterance, however, the accuracy of the tongue point $y$-coordinate estimates is reasonably high.

The inertia of speech articulators means they are constrained to move relatively slowly and smoothly. While continuity is a basic principle of human speech production, an MLP trained with the sum-of-squares error function will not necessarily emulate this. The aim of training with the sum-of-squares error function is to minimise the distance of the MLP output from the target output for each input–output training pattern. In other words, the optimal instantaneous mapping that the MLP can provide at a given time frame is the conditional average of the articulator configurations for *all* such input vector frames. However, this optimum does not in any way stipulate that the output articulatory configuration at one time frame should depend on the articulatory configuration at any previous or following time frames. Therefore, we rely heavily on the inversion mapping function itself to yield smoothly varying output. Specifically, we assume that acoustic input vectors at time $t-1$ and time $t+1$ will map to points in the near vicinity of the articulatory configuration at time $t$, such that no discontinuities result in the overall sequence of MLP output. This assumption is reasonable, but is in practice liable to be confounded by one-to-many mappings from the acoustic to the articulatory domain. For instance, the MLP output shown in Fig. 2 appears to be noisier than the respective measured articulatory trajectories. This is despite the use of a context window, which should help alleviate the problem of one-to-many mappings.

In an effort to make up for inadequacies in the MLP estimated articulatory trajectories, we might turn to various postprocessing techniques. For example, we know that the articulators move relatively slowly. Therefore, one simple example of an articulatory constraint we could impose as a postprocessing step, which utilises this knowledge, is to lowpass filter the MLP output. In Richmond (2002), we have demonstrated that lowpass filtering the MLP estimated trajectories, with channel specific lowpass cutoff frequencies which had been identified empirically, does indeed moderately reduce RMS error and increase correlation scores. However, we will not use this method here. In the present paper, we want simply to evaluate whether a more sophisticated description of the target articulatory domain might provide a better model.

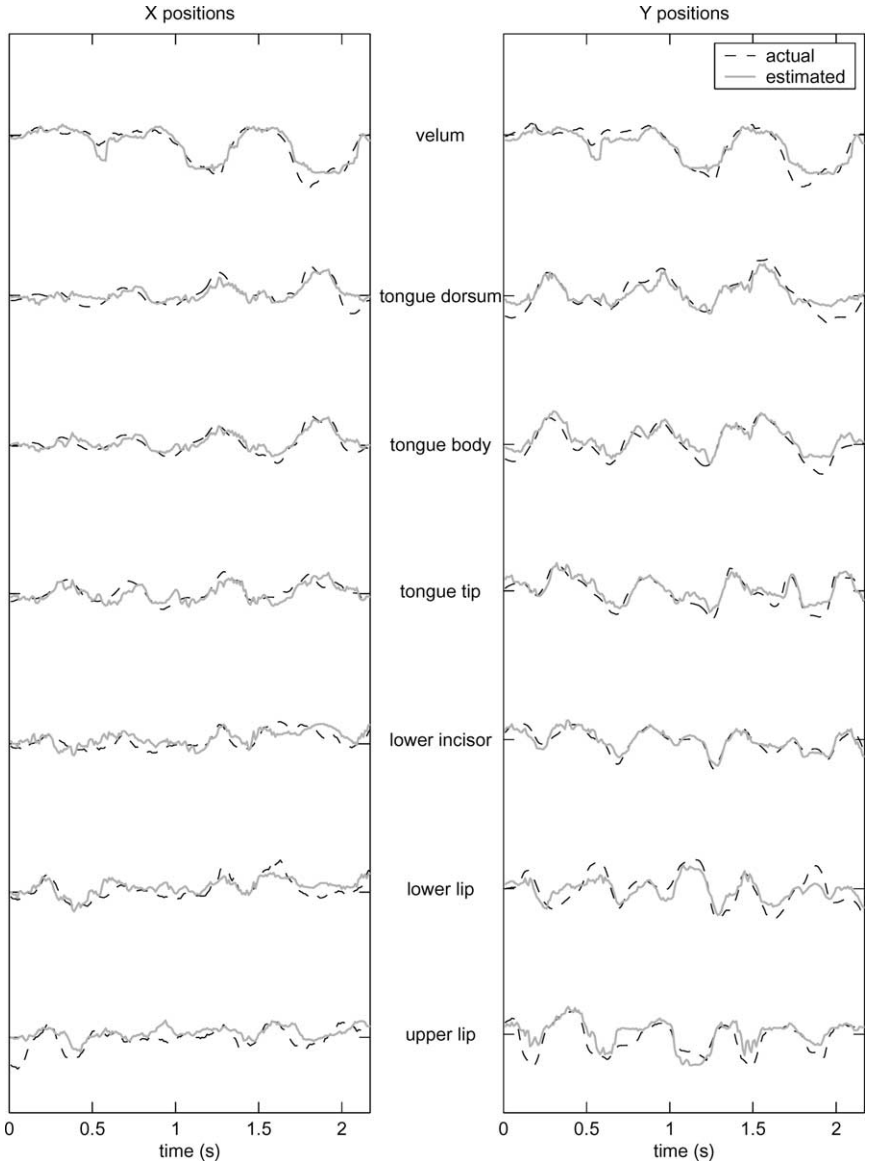


Fig. 2. A comparison of the MLP estimated articulatory trajectories with the EMA-measured articulatory trajectories for the unseen test utterance "The speech symposium might begin on Monday". Note the articulatory trajectories in this plot are shown in their normalised range, after processing as described in Section 2.2.

## 4. MDN inversion mapping

To recapitulate, the MDN provides a principled method for modelling the target data corresponding to each input vector with a full conditional probability density function. This contrasts with the least-squares trained MLP, which only provides a mean value and a fixed residual

variance for the distribution of possible target values. Moreover, the MLP assumes distributions of target points corresponding to a unimodal Gaussian.

Since MDNs are not commonplace in the speech field, for the reader's convenience we shall first briefly introduce the theory underpinning the model. For a complete description, the reader is directed to Bishop (1994) or Bishop (1995).

### 4.1. Introduction to the mixture density network

An MDN can be thought of as the combination of a conventional neural network with a mixture model. An example MDN is shown in Fig. 3. In this example, the MDN takes an input vector $\mathbf{x}$ of dimensionality 5 and gives the conditional probability density of a vector $\mathbf{t}$ of dimensionality 1 in the target domain. This density function is modelled by a Gaussian mixture model with three components, so that it is given by:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{M} \alpha_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}), \tag{3}$$

where $M$ is the number of mixture components (in this example, 3), $\phi_j(\mathbf{t}|\mathbf{x})$ is the conditional probability density given by the $j$th kernel, and $\alpha_j(\mathbf{x})$ is the mixing coefficient for the $j$th kernel. The mixing coefficients represent the *prior* probability that a target vector $\mathbf{t}$ has been generated by the $j$th kernel, and therefore they must sum to one. Note that any of a number of different kernel functions may be used in the mixture model, but only Gaussian kernel functions are considered here. In theory, any neural network with universal approximation capabilities can be used to map

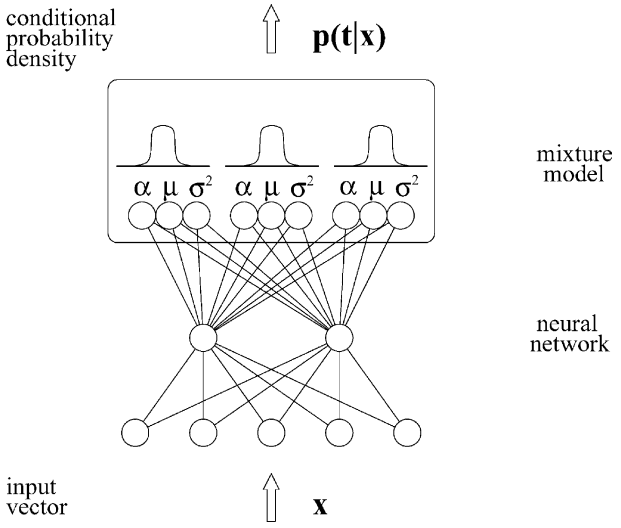


Fig. 3. The mixture density network is the combination of a mixture model and a neural network. In a trained MDN, the neural network maps from the input vector $\mathbf{x}$ to the control parameters of the mixture model, which in this case uses Gaussian components (priors $\alpha$, means $\mu$ and variances $\sigma^2$) but in theory could be any of a number of kernel functions. The mixture model gives a full pdf description of the target domain conditioned on the input vector $p(\mathbf{t}|\mathbf{x})$.

from the input vector to the mixture model parameters. In this example, we see a feedforward MLP with 5 input units, a hidden layer of 2 units with sigmoidal activation and nine linear output units for the mixture parameters. In general, the total number of network outputs is given by $(c+2) \times M$, where $c$ is the dimensionality of the target domain, and $M$ is the number of mixture components. In other words, each mixture component has 1 unit for its prior, 1 unit for its variance and $c$ units for the mean of the component in the target space. Note that we are using Gaussian components with spherical covariance here (one variance parameter for each component). In principle, the MDN is not limited to using only spherical covariance; both a diagonal or full covariance matrix could be used for each component. However, complicating the model in this way is avoidable, because a mixture of spherical Gaussians is theoretically able to model any distribution function with arbitrary accuracy assuming enough components are available (Bishop, 1995).

In order to constrain the mixing coefficients to lie within the range $0 \leqslant \alpha_j(\mathbf{x}) \leqslant 1$ and to sum to unity, the *softmax* function (Bridle, 1990) is used to relate the mixing coefficients of the mixture model to the output of the corresponding units in the neural network:

$$\alpha_j = \frac{\exp(z_j^\alpha)}{\sum_{l=1}^{M} \exp(z_l^\alpha)}, \tag{4}$$

where $z_j^\alpha$ is the output of the neural network corresponding to the mixture coefficient for the $j$th mixture component. The variance parameters of the mixture model are related to the corresponding outputs of the neural network according to the following function:

$$\sigma_j = \exp(z_j^\sigma), \tag{5}$$

where $z_j^\sigma$ is the output of the neural network corresponding to the variance for the $j$th mixture component. This has the convenience of avoiding the variance becoming less than or equal to zero. Finally, the means for the mixture model are represented directly by the corresponding outputs of the neural network:

$$\mu_{jk} = z_{jk}^\mu, \tag{6}$$

where $z_{jk}^\mu$ is the value of the output unit corresponding to the $k$th dimension of the mean vector for the $j$th mixture component.

The objective of training the MDN is to minimise the negative log likelihood of the observed target data points given the mixture model parameters, which is given by

$$E = -\sum_n \ln \left\{ \sum_{j=1}^{M} \alpha_j(\mathbf{x}^n) \phi_j(\mathbf{t}^n | \mathbf{x}^n) \right\}. \tag{7}$$

Since it is the neural network part of the MDN that provides the parameters for the mixture model for each input–output vector training pair, this error function must be minimised with respect to the network weights. Fortunately, the derivatives of the error at the network output units corresponding separately to the priors, means and variances of the mixture model may be calculated (see Bishop, 1995). These error 'signals' may then be propagated back through the network as normal in network training to find the derivatives of the error with respect to the network weights. Thus, training is a problem to which standard non-linear optimisation algorithms can be applied.

### 4.2. Application to inversion mapping task

Using a single Gaussian in the mixture model is similar to the unimodal regression observed when using an MLP to estimate articulation from acoustics. However, in the case of the MDN, the variance of the Gaussian as well as the mean is allowed to vary as a function of the acoustic input vector, which yields greater modelling flexibility. Using a mixture of Gaussians in the MDN pdf (see Eq. (3)) introduces the additional flexibility of allowing potentially non-Gaussian and multimodal distributions in the target articulatory domain to be modelled more closely.

In Richmond (2002), we compared the performance of MDNs trained for each articulatory channel containing four combinations of either 5 or 10 hidden units and either one or two Gaussian kernels in the density model. In 27 out of 28 cases where networks were directly compared, the MDN with a mixture of two Gaussians in the density model demonstrated higher accuracy than the equivalent MDN with a single Gaussian. The dependence on the number of hidden units was also indicated by a pairwise comparison of otherwise equivalent networks. In 27 [1] out of 28 direct comparisons, the MDN containing 10 hidden units performed better than the equivalent MDN with just 5 hidden units. For the sake of brevity, we will concentrate here on the results for the MDNs containing 10 hidden units and 2 Gaussians in the density function for each articulatory channel.

This limited set of combinations of hidden layer size and density function may not include the optimal MDN architecture. However, our aim in the current paper is simply to ascertain whether having variance parameters dependent on the input on one hand and allowing multimodal distributions on the other can provide a closer model of the distribution of possible target articulatory configurations. Since our results will indicate this is the case, this limited set is sufficient for our purposes here. Nevertheless, there are strong indications that it may prove beneficial in future work to try MDNs with higher numbers of hidden units and mixture components in order to gain better performance.

Exactly the same training, validation and testing data sets as were used with the MLP described in Section 3 were used for training the MDNs. Thus, the MDNs in this section had 400 acoustic input units, which made up the context window of 20 frames of 20 filterbank coefficients.

Prior to training, the biases for the output units of the MDN were initialised using a $k$-means based initialisation algorithm. Under this algorithm, a separate Gaussian mixture model of the same form as the MDN output was used to model the *unconditional* density of the target data. Ten iterations of the $k$-means algorithm were used to determine the component centres. The priors were computed from the proportion of the target data belonging to each component, and the variances were calculated as the sample variance of the target data points belonging to each component from the associated mean. The output unit biases were then set so that the net would output the values in the Gaussian mixture model. All other biases and weights in the MDN were randomised by sampling from a Gaussian.

[1] Not the same 27 networks from the previous comparison between one and two Gaussian kernels.

As with the MLP, the MDNs were trained using the Scaled Conjugate Gradient algorithm to optimise the error function given in Eq. (7). Although the separate validation set was used to identify the best network during training, an upper bound of 2000 training epochs was also imposed. However, all 14 networks appeared to reach a maximum in terms of performance on the validation set within this limit.

### 4.3. Results

Figs. 4 and 5 provide a visual representation of the output of the MDNs. These plots, which might be termed articulatory "probabilitygrams", show the probability density for the location of an articulator within its range of movement as a function of time. Such plots are produced by taking the probability density function output by the MDN at each time frame ($x$-axis) and calculating the probability density at certain intervals (in this case 0.01) encompassing the range of movement of the given variable ($y$-axis). The probability density for a particular articulator location at a given time frame is represented by the greyscale intensity. Intense black indicates high probability density, whereas white indicates low probability density.

As well as showing the output of the MDNs, the measured EMA trajectories are shown for comparison. Phonetic segmentation is also indicated in these plots. This labelling is provided as standard as part of the MOCHA database, and was produced automatically using the recording prompts and an HMM forced alignment.

#### 4.3.1. Variability of variance

In Figs. 4 and 5, it is clear that the variance around the estimated location of an articulator is higher at some times than at others. We also notice a correlation: when the MDN output variance is low, the accuracy of the estimated location of the articulator is typically higher. This correlation

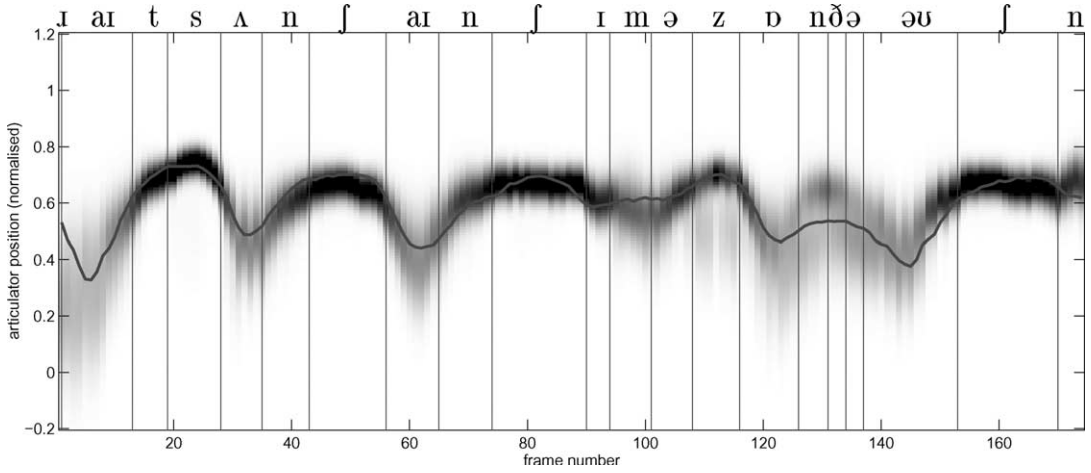


Fig. 4. Comparison of MDN output with the actual, measured articulator trajectory for the unseen test utterance "Bright sunshine shimmers on the ocean". The $y$-coordinate trajectory of the lower incisor is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density.

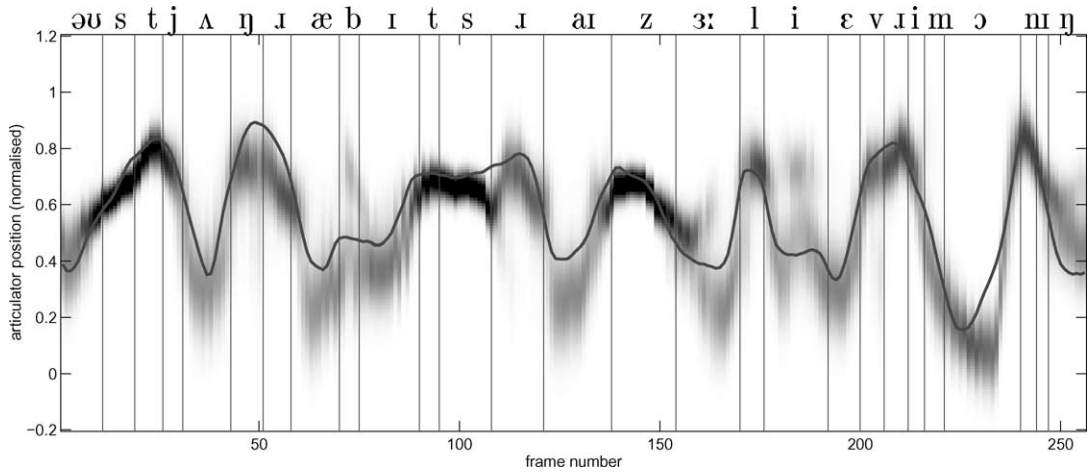


Fig. 5. Comparison of MDN output with the actual, measured articulator trajectory for the unseen test set utterance "Most young rabbits rise early every morning". The *y*-coordinate trajectory of the tongue tip is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density.

is unsurprising, since the variances of the probability density functions output by the MDN are estimated as a function of the acoustic input so as to minimise the negative log likelihood error function.

Generally, it seems that the variance is low when the articulators might be critical to the production of the segment. For example, for the tongue tip articulator in Fig. 5, the variance is lowest during times when the tongue tip is near the upper surface of the vocal tract and "critical", such as during the phones [t, s, z]. Table 2 explores this observation in greater depth.

For each of the three articulatory channels in Table 2, we have calculated the average variance output by the respective MDN during all frames in the test set corresponding to 23 separate phones. [2] Channel specific MDNs containing 10 units in a single hidden layer and a single Gaussian kernel in the output pdf were used for this purpose (Richmond, 2002). For each articulatory channel, we have then ranked the phones according to their respective MDN average variance.

On the whole, the results in Table 2 are consistent with our initial observation. For example, we find that for the "tongue tip *y*" channel, those phones for which the position of the tongue tip is critical demonstrate lower average variance, such as [s, z, θ...]. Meanwhile, for the "upper lip *x*" channel, phones such as [m, w, b, p] demonstrate lower average variance relative to other phones. In the third column, for the "velum *x*" channel, it is interesting to note that the MDN output features lower average variance for those phones where the velar port is most likely to be closed, whereas for the nasal stops [m, n, ŋ] the average variance is relatively high.

[2] The remaining 23 phones in the phone set that have not been included in Table 2 were all vowels.

Table 2
Ranked phone dependent averages for the variance parameters of MDN output density functions for three articulatory channels

| Tongue tip $y$ | | Upper lip $x$ | | Velum $x$ | |
|---|---|---|---|---|---|
| Phone | Av. $\sigma^2$ | Phone | Av. $\sigma^2$ | Phone | Av. $\sigma^2$ |
| s | 0.003 | m | 0.015 | ʧ | 0.007 |
| z | 0.003 | w | 0.016 | z | 0.007 |
| θ | 0.005 | ð | 0.017 | b | 0.008 |
| ʃ | 0.006 | n | 0.018 | ʃ | 0.008 |
| t | 0.009 | b | 0.019 | s | 0.008 |
| ð | 0.010 | p | 0.019 | g | 0.009 |
| ʧ | 0.011 | ŋ | 0.019 | ʤ | 0.009 |
| n | 0.011 | k | 0.020 | t | 0.009 |
| y | 0.012 | f | 0.020 | y | 0.009 |
| ŋ | 0.012 | ɹ | 0.020 | d | 0.010 |
| d | 0.012 | g | 0.020 | k | 0.011 |
| ɹ | 0.012 | l | 0.021 | f | 0.011 |
| ʤ | 0.013 | v | 0.021 | θ | 0.011 |
| v | 0.013 | h | 0.021 | w | 0.012 |
| f | 0.013 | d | 0.021 | p | 0.012 |
| l | 0.014 | θ | 0.021 | v | 0.012 |
| m | 0.014 | z | 0.022 | l | 0.013 |
| h | 0.014 | t | 0.024 | ð | 0.013 |
| g | 0.014 | s | 0.027 | ɹ | 0.018 |
| k | 0.015 | ʤ | 0.027 | ŋ | 0.021 |
| w | 0.016 | ʧ | 0.028 | m | 0.024 |
| p | 0.017 | ʃ | 0.028 | h | 0.025 |
| b | 0.017 | y | 0.029 | n | 0.037 |

Phone identities are given using standard IPA notation. Notice that the MDN output generally features lower variance where an articulator's position could be considered to exert a strong influence during the production of a phone.

There are of course certain caveats to bear in mind when considering the results in Table 2. From a practical point of view, it is possible that the picture is clouded to some extent by the use of the automatically produced MOCHA labelling, which in some respects is closer to a "phonological" labelling rather than a fine-grained "phonetic" labelling: for example, any instances of glottal stops will be labelled as the underlying phone; instances of assimilation are overlooked; no distinction is made between dark and light [l], and so on. In other words, the phone classes used may not correspond to satisfactorily pure groups of allophones. It is also possible, hypothetically speaking, that an articulator's position may not be critical to the production of a given phone, but that its position may nevertheless be relatively accurately predicted (perhaps under the influence of a critical articulator). Finally, when interpreting these results, it is important not to forget that the variances in question are not those exhibited directly by the articulators themselves. Instead, they are the variance parameters from the MDN output pdfs, which are only an *approximation* to the conditional probability density of the real data. However, despite these caveats, we nevertheless believe these findings tend to support the view of critical versus non-critical articulators discussed by Papcun et al. (1992).

### 4.4. MDN output flexibility

The probability density function over the target domain which the MDN provides is a flexible model in itself and may be used in numerous ways according to requirement. For example, we could compute the conditional mean of the target data given the input vector. This value approximates the output of a standard least-squares trained MLP as a special case. We could also compute the variance around this average for each input vector. This goes beyond the capabilities of a standard MLP, for which we can only compute a global residual variance. On the other hand, at each time frame, we could take the mean and variance of the Gaussian with the highest prior to give the mode trajectory. This approximates the mixture of experts model (Jordan & Jacobs, 1994) as another special case. While these few examples give an impression of the flexibility of the MDN output, many other ways of using the MDN output may be envisaged.

## 5. Comparing MLP with MDN

In order to gauge whether MDNs allow closer modelling of the distributions of possible articulatory configurations than MLPs, we require an error measure to compare like with like. If we assume that the MLP has sufficient representational power and has been trained well enough to approximate closely the conditional mean of the target data, we know that any remaining error is attributable to the variance of the target data itself around its conditional mean (Bishop, 1995).

The global variance for each articulatory channel ($\sigma_k^2$) may be calculated as

$$\sigma_k^2 = \frac{1}{N}\sum_{n=1}^{N}\left\{y_k(\mathbf{x}^n;\mathbf{w}^*) - t_k^n\right\}^2, \tag{8}$$

where $N$ is the number of input–output vector pairs in the training set, $y_k(\mathbf{x}^n;\mathbf{w}^*)$ is the $k$th output of the trained MLP with the optimal weight configuration $\mathbf{w}^*$ in response to the acoustic input vector $\mathbf{x}^n$, and $t_k^n$ is the corresponding articulatory target value for channel $k$.

Hence, at testing time, we can interpret the MLP output for each time frame as a single Gaussian probability density function, whose mean and variance are given respectively by the output of the MLP and the global variance precomputed according to Eq. (8). Given the probability density functions at each time frame, we can calculate the likelihood of the articulatory target data for the whole test set. This likelihood can be compared directly with the equivalent likelihood calculated according to the probability density functions provided by the MDNs.

Table 3 provides just this comparison. Columns 2 and 3 of Table 3 give the geometric mean likelihood of the articulatory target data in the test set, [3] given the framewise probability density functions computed from the MLP and MDN, respectively. This was calculated by dividing the total log likelihood by the number of input–output pairs in the test set, then transferring back out of the log domain. In column 4 we have calculated the relative improvement of the MDN model over the MLP in terms of these likelihoods as a percentage.

[3] There were 11,660 acoustic-articulatory feature vector pairs in the test set.

Table 3
A comparison of the likelihood of the test set target data given the framewise probability density functions provided by MLP and MDN

| Articulator channel | MLP mean likelihood | MDN mean likelihood | Increase % |
|---|---|---|---|
| Upper lip $x$ | 1.35 | 1.40 | 4.4 |
| Upper lip $y$ | 1.69 | 1.83 | 8.2 |
| Lower lip $x$ | 1.52 | 1.56 | 2.6 |
| Lower lip $y$ | 1.75 | 1.88 | 7.4 |
| Lower incisor $x$ | 1.49 | 1.57 | 5.3 |
| Lower incisor $y$ | 2.04 | 2.48 | 21.8 |
| Tongue tip $x$ | 1.90 | 2.01 | 5.4 |
| Tongue tip $y$ | 2.11 | 2.56 | 21.4 |
| Tongue body $x$ | 1.97 | 2.08 | 5.4 |
| Tongue body $y$ | 2.18 | 2.38 | 9.0 |
| Tongue dorsum $x$ | 1.92 | 2.02 | 5.6 |
| Tongue dorsum $y$ | 1.79 | 2.00 | 11.9 |
| Velum $x$ | 1.90 | 2.14 | 12.7 |
| Velum $y$ | 1.88 | 2.00 | 6.7 |

Here, 14 channel specific MDNs have been used with 10 units in one hidden layer and an output mixture density model comprising two Gaussian kernels. The likelihood figures for both MLP and MDN are given as the geometric mean, calculated over all input–output pairs in the unseen test set.

To reiterate Section 1.2, in performing this comparison, we are effectively asking whether there is any point in attempting to model the distribution of possible articulatory points conditioned on the input vector, and whether using distributions more complex than a single Gaussian can also provide a better model. If the residual error in the MLP inversion mapping were due to machine measurement error, or any other source of error that was not dependent on the acoustic signal, then we would not expect to see any benefit in using the MDN. However, if the distribution of target articulatory points does depend to any extent on the acoustic input, and thus is characteristic of the inversion mapping function itself, then the MDN should demonstrate a higher likelihood score.

As the last column of Table 3 makes clear, the use of MDNs does indeed yield higher likelihood scores for all articulatory channels. The size of this improvement ranges from 2.6% to 21.8%. Apart from for the velum, it would appear that the best improvement is observed for the $y$-coordinates of the articulators. Excluding the velum channels, the average improvement for the articulatory-coordinates is about 13.3%, whereas the average for the $x$-coordinates is only 4.8%. In future work, it would be interesting to investigate what might be the reason for this disparity. For example, it may be this is merely coincidence, or an idiosyncrasy of this particular speaker. On the other hand, it could be a more general observation. Experiments using data from multiple speakers would be useful to investigate this question.

### 5.1. Density function form

As mentioned in Section 4.2, we have found that MDNs with 2 Gaussian kernels outperformed MDNs with a single Gaussian on the same inversion mapping task in 27 out of 28 cases of comparison. This indicates that the reason the MDNs outperform the MLP in Table 3 is at least in part due to greater flexibility for modelling non-Gaussian distributions.

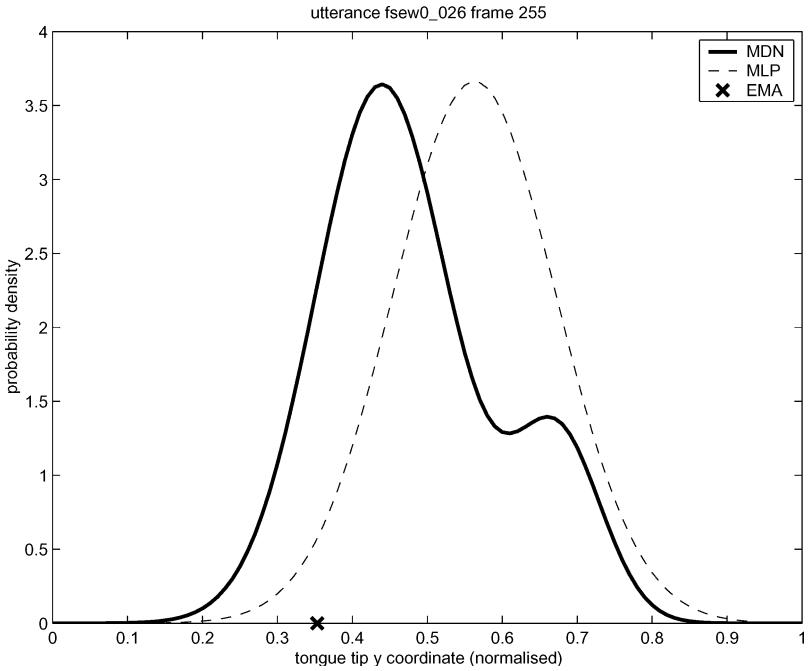


Fig. 6. The probability density function output by the MDN (containing 10 hidden units and a mixure density model of two Gaussian components) for the *y*-coordinate of the tongue tip at frame 255 (/ŋ/ phone) of utterance `fsew0_026` (as shown in Fig. 5). The conditional density function output by the MDN at this point is clearly not consistent with a unimodel Gaussian. For comparison, we also include the equivalent density function for the MLP output, as well as the *y*-coordinate of the tongue provided by EMA.

Fig. 6 illustrates this point. This figure shows the density function output by the MDN for the *y*-coordinate of the tongue tip at frame 255 of utterance `fsew0_026` (as shown in Fig. 5). This conditional density function is clearly different from a unimodal Gaussian. We have also marked the actual articulator location on this plot with an "*x*", and overlaid a unimodal Gaussian whose mean is given by the MLP output and whose variance is the residual variance calculated according to Eq. (8). If we assume the probability density function provided by the MDN is a reasonable approximation to the real conditional distribution of the target data, we see that by accommodating the probability mass under the "lobe" on the right hand side (apparently centred just below 0.7), the accuracy of the MLP estimate of the articulator position has been compromised.

## 6. Conclusion

A feedforward MLP has been presented and applied to the task of recovering articulatory trajectories from acoustics during continuous, phonetically rich speech. This network was trained and tested using measured articulatory-acoustic data.

While the performance of this network compared favourably with the results of other inversion methods reported in the field, we suspected that the MLP was ill-suited to estimating articulation from acoustics in at least two respects. First, the MLP is limited to modelling target data points

with a unimodal Gaussian, which is not necessarily a sound assumption to make when modelling the inversion mapping. Second, the MLP gives no indication of the variance of the distribution of the target points around the conditional average.

In theory, the MDN is able to provide a description or model of the target domain at each time frame which is powerful enough to overcome these shortfalls. In order to verify this, we have sought to compare directly the performance of the MDN with that of the MLP on the same inversion task.

By reinterpreting the MLP output error in terms of the likelihood of the target data, we have demonstrated that the MDN does indeed provide a more accurate model of the distribution of the target variables in the articulatory domain with respect to the inversion mapping problem. This result is relevant not only to those using MLPs in particular to perform the inversion mapping, but wherever similar assumptions about the conditional distribution of articulatory parameters are made.

Furthermore, we have indicated evidence that might explain how this improvement is realised. Specifically, we have shown that the MDN is able to accommodate the fact that the positions of the articulators may be more constrained, and thus more reliably estimated, at some times than at others. In addition, the MDN is better able to handle cases where the conditional distribution of articulator positions does not correspond in form to a unimodal Gaussian, for example where the solution to the inversion mapping has multiple branches.

## 7. Future work

This paper has concentrated on demonstrating that the MDN is able to provide a more flexible and accurate model of the articulatory domain when attempting to infer articulatory parameters from the acoustic speech signal. However, we have not considered how the output of an MDN might best be used in applications. Although we touched briefly on a few values that might be computed from the pdf output by the MDN in Section 4.4, we have by no means exhausted the scope of possibilities.

Future work will therefore focus specifically on exploring how best to exploit the MDN output. The aim will be to derive the "best-guess" trajectory through the sequence of density functions. For example, one method envisaged would be to use Kalman smoothing, where the variance output by the MDN is used as an estimate of the measurement error of the articulator position observations. Articulatory constraints of varying degrees of complexity could be employed within this technique. Perhaps the simplest would be to constrain the articulators to move slowly from one time frame to the next. More sophisticated articulatory constraints could be instantiated by following the approach of Dusan (2000), who used trained Kalman filter parameters specific to the transition between all phones in the phone inventory.

## Acknowledgements

The authors would like to thank Steve Isard for contributing many valuable comments and suggestions in response to previous drafts, and Alan Wrench for help with the MOCHA data.

## References

Atal, B.S., Chang, J.J., Mathews, M.V., Tukey, J.W., 1978. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer sorting technique. J. Acoust. Soc. Am. 63, 1535–1555.

Bishop, C., 1994. Mixture density networks. Technical Report NCRG/4288, Neural Computing Research Group, Department of Computer Science, Aston University, Birmingham, B4 7ET, UK, February.

Bishop, C., 1995. In: Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Bridle, J.S., 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Neurocomputing: Algorithms, Architectures and Applications. Springer, Berlin, pp. 227–236.

Dusan, S., 2000. Statistical estimation of articulatory trajectories from the speech signal using dynamical and phonological constraints. Ph.D. Thesis, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, April.

Hogden, J., Lofqvist, A., Gracco, V., Zlokarnik, I., Rubin, P., Saltzman, E., 1996. Accurate recovery of articulator positions from acoustics: new conclusions based on human data. J. Acoust. Soc. Am. 100 (3), 1819–1834.

Jordan, M., Jacobs, R., 1994. Hierarchical mixtures of experts and the EM algorithm. Neural Comput. 6 (2), 181–214.

Lindblom, B., Lubker, J., Gay, T., 1979. Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation. J. Phonet. 7, 147–161.

Mermelstein, P., 1973. Articulatory model for the study of speech production. J. Acoust. Soc. Am. 53 (4), 1070–1082.

Møller, M., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6 (4), 525–533.

Mozer, M.C., Smolensky, P., 1989. Skeletonization: a technique for trimming the fat from a network via relevance assessment. In: Touretzky, D.S. (Ed.), Advances in Neural Information Processing Systems, vol. 1. Morgan Kaufmann, Los Altos, CA, pp. 107–115.

Okadome, T., Suzuki, S., Honda, M., 2000. Recovery of articulatory movements from acoustics with phonemic information. In: Proc. 5th Seminar on Speech Production. Kloster Seeon, Bavaria, pp. 229–232.

Papcun, G., Hochberg, J., Thomas, T.R., Laroche, F., Zachs, J., Levy, S., 1992. Inferring articulation and recognising gestures from acoustics with a neural network trained on X-ray microbeam data. J. Acoust. Soc. Am. 92 (2), 688–700.

Rahim, M., Goodyear, C., Kleijn, W., Schroeter, J., Sondhi, M., 1993. On the use of neural networks in articulatory speech synthesis. J. Acoust. Soc. Am. 93 (2), 1109–1121.

Rahim, M.G., Kleijn, W.B., Schroeter, J., Goodyear, C.C., 1991. Acoustic-to-articulatory parameter mapping using an assembly of neural networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 485–488.

Richmond, K., 2002. Estimating articulatory parameters from the acoustic speech signal. Ph.D. Thesis, The Centre for Speech Technology Research, Edinburgh University.

Roweis, S., 1999. Data driven production models for speech processing. Ph.D. Thesis, California Institute of Technology, Pasadena, California.

Shirai, K., Kobayashi, T., 1986. Estimating articulatory motion from speech wave. Speech Commun. 5, 159–170.

Wakita, H., 1979. Estimation of vocal-tract shapes from acoustical analysis of the speech wave: the state of the art. IEEE Trans. Acoust. Speech Signal Process. ASSP-27, 281–285.

Westbury, J.R., 1994. X-ray microbeam speech production database user's handbook. Technical Report Version 1.0, University of Wisconsin, Madison.

Wrench, A., Hardcastle, W.J., 2000. A multichannel articulatory speech database and its application for automatic speech recognition. In: Proc. 5th Seminar on Speech Production. Kloster Seeon, Bavaria, pp. 305–308.

Zachs, J., Thomas, T.R., 1994. A new neural network for articulatory speech recognition and its application to vowel identification. Computer Speech Language 8, 189–209.