MIXTURE DENSITY NETWORKS, HUMAN ARTICULATORY DATA AND ACOUSTIC-TO-ARTICULATORY INVERSION OF CONTINUOUS SPEECH

K. Richmond   Centre for Speech Technology Research, University of Edinburgh, UK

1 INTRODUCTION

Researchers have been investigating methods for retrieving the articulation underlying an acoustic speech signal for more than three decades. A successful method would find many applications, for example: low bit-rate speech coding, helping individuals with speech and hearing disorders by providing visual feedback during speech training, and the possibility of improved automatic speech recognition.

A good deal of work has been based on analytical techniques, such as inverse filtering, or on the use of articulatory synthesis models. However, thanks to technologies such as as X-Ray microbeam (XRMB) cinematography and electromagnetic articulography (EMA), measured human articulatory data has become increasingly accessible. This real, human data is arguably preferable to the use of analytical or articulatory synthesis models, where intrinsic flaws in the models themselves can confuse the picture.

A relatively small number of empirical learning models applied to human articulatory data have been described in the literature. These include extended Kalman filtering ([5]), artificial neural networks[1] ([14]), self-organising HMMs ([16]) and codebook methods ([7]). However, these efforts have mostly been limited to some subsection of full speech, such as a few stop-consonants or vowel transitions.

In previous work at CSTR, we have investigated using both feed-forward and recurrent neural networks to model the inversion mapping during full, continuous speech. In such network models, where the aim of training is to minimise a sum-of-squares error function, the outputs approximate the average of the target data, conditioned on the input vector. However, this conditional average can provide only a very limited, and in many cases problematic, description of the variables in the target domain. This limitation is particularly acute for ill-posed problems where the mapping may be multi-valued, such as in the case of the acoustic-to-articulatory inversion mapping. For this task, the average of several correct target values is not necessarily itself correct.

Mixture density networks (MDN) represent a principled method to model full probability density functions over the positions of the target variables in the articulatory domain, conditioned on acoustic input vectors. This is achieved by modelling the conditional distribution for each target vector with

[1] Hereafter referred to simply as "neural networks" or ANNs

a Gaussian mixture model, the parameters for which are generated using a multilayer perceptron (MLP) trained using a standard optimisation technique, but with specialised error functions.

We report here on applying MDNs to acoustic-to-articulatory inversion. The data used for the experiments we describe comprise 460 (British) TIMIT sentences read by a female speaker, which were recorded at the EMA facility located at Queen Margaret College, Edinburgh. This corpus is designed to capture wide phonetic diversity, and hence maximises the potential for investigating the one-to-many problem inherent in the inversion mapping.

This paper will first provide a brief introduction to the difficulties inherent in the acoustic-to-articulatory inversion mapping. Specifically, we shall look at *non-uniqueness*. Next, we introduce the human articulatory data that has been used in the experiments described, and how it was processed for use as training data for the neural networks. The characteristics of two networks for performing the acoustic-to-articulatory inversion mapping are then compared. The first of these is a standard feedforward multi-layer perceptron (MLP). The second is a mixture density network. As MDNs are not as well known as the more common neural network models, we include a short description of the model itself for the reader's convenience.

2 NON-UNIQUENESS

Human experimental evidence from as early as the 1920's has demonstrated that a speaker is capable of producing acoustic signals very close to the intended sound even when the jaw is fixed in an unnatural position by a bite-block. For example, [11] compared the formant frequencies for four Swedish vowels (/i,u,o,a/) produced by six speakers both when the jaw was free to move and when fixed in a certain position by a bite block (2.5 and 22.5mm thick). They found that despite the physiologically unnatural position of the jaw enforced by the bite block, the speakers were able to produce vowels with formant patterns well within the range of variation of a set of vowels spoken under normal conditions. Moreover, they comment that to achieve this, the speakers did not require any training time or practice. The capability of a speaker to vary their articulation in order to produce the desired speech sound has been termed *articulatory compensation*. As an extreme, if somewhat anecdotal, example of articulatory compensation, consider the ventriloquist, who can produce an intelligible speech stream while appearing to not to be articulating anything!

The existence of articulatory compensation suggests that a given acoustic speech frame may have been produced by any of a number of articulatory configurations. In fact, a significant amount of evidence has accrued from other research areas which supports this view of speech production, for example mathematical theory, manipulation of articulatory synthesis models and analysis of human articulatory data.

Work done using articulatory synthesis models has raised several doubts concerning the overall viability of acoustic-to-articulatory inversion. For example, [1] placed significant emphasis on investigating what they called "fibers" in the articulatory space, and applied a numerical inversion method to studying them. A fiber is defined as a region in articulatory space within which movement produces no change in the corresponding acoustic output. They demonstrated, for example, that the mouth opening of their articulatory model can vary considerably without affecting the formants characteristic of an /i/ vowel. As an additional example, they presented multiple vocal tract area functions that have identical first three formant frequencies for several vowels.

Although bite block experiments and articulatory synthesis models indicate **in theory** that multiple configurations of the articulators are capable of producing a single speech sound, it is not necessarily the case that this should occur in **normal human speech**. For this reason, the empirical evidence put forward by Roweis ([16]) is most compelling. Roweis had at his disposal a database of approximately 175,000 midsagittal articulatory "snapshots", along with the simultaneous audio signal, recorded by X-ray microbeam cinematography at Wisconsin University. From this he compiled a data set of acoustic-articulatory vector pairs. The acoustic feature vectors consisted of line spectral pairs (LSP) calculated over a 23.5ms window centred on each articulatory sample time point. The articulatory feature vector comprised the x- an y- coordinates of eight articulator points.

One of these acoustic-articulatory vector pairs was then selected and its 1000 nearest neighbouring vector pairs in the acoustic domain were found. This was done using a Mahalanobis distance metric based on the global covariance of the acoustic data. The vector pairs found in this way could be plotted in the articulatory domain (i.e. the x- coordinate against the y- coordinate separately for each of the eight articulatory points) to produce scatter plots. If the points in articulatory space fell within a tightly constrained area, it would mean the that the relationship between acoustics and articulation is a straightforward mapping. However, the plots produced in [16] feature wide spreads and multimodal distributions for the points in articulatory space corresponding to neighbouring points in acoustic space.

If humans do use a range of articulations to produce a single given acoustic signal, then the inversion mapping is a classic example of what is termed an *ill-posed* problem, as the solution may be non-unique. It raises the question of how an inversion algorithm should decide between all the possible configurations that might be associated with a given acoustic feature vector.

Among all the different approaches to performing the inversion mapping described in the literature, we can identify a common strategy for attempting to disambiguate one-to-many mappings:

1. perform an instantaneous mapping from the acoustic domain to the articulatory domain
2. perform some sort of post processing or smoothing on the resulting trajectories on the basis of some articulatory constraints.

In other words, researchers have sought ways to use the continuous and relatively slow movements of the articulators to disambiguate instantaneous uncertainty. This is done by incorporating additional constraints on the articulatory configurations recovered from acoustics and their dynamic behaviour from one time frame to the next. However, within this common strategy of using articulatory constraints, there is a dichotomy in philosophy. On one hand, some researchers choose to ignore instantaneous non-uniqueness in the hope that errors introduced by doing so will be minimised by the continuity constraints. On the other hand, researchers take instantaneous non-uniqueness into account and build into their model some means for dealing with multiple candidate output articulatory configurations for each acoustic input time frame. In this case, the intention is that articulatory constraints can chart some optimum course through the series of possibilities; correct information pertaining to articulator positions recovered when the mapping is not ill-posed when taken with constraints on articulatory movements will disambiguate problematic sections of speech where the mapping is non-unique.

The constraints used can vary widely in complexity. Lowpass filtering the articulatory trajectories imposes an articulatory constraint of sorts when the pass band is set to equal the bandwidth observed for human articulatory movements. A whole range of more complex articulatory constraints have also been employed. One example of the constraints used is [15], who as part of a dynamic programming search through the output of their networks used a cost function constraining articulatory trajectories to be a smooth as possible. Meanwhile, others, e.g. [10] suggest employing the constraint of economy of effort. The ideal trajectory under this constraint is one where the articulators move as little as possible. More elaborate techniques have been suggested, such as [4], whose system features a recurrent algorithm that takes into account the dynamic properties of the articulators. At time t, the position of the articulators at time $t + 1$ is forward estimated using the current position together with the velocity and acceleration of the articulatory parameters. Then, at time $t + 1$, the candidate positions for the articulators given the acoustics (they use a codebook approach in this case) are compared to the estimate calculated at the previous time step, and the best is selected.

Where researchers have turned to articulatory constraints in an effort to circumvent instantaneous non-uniqueness, the underlying motivation generally seems to be to recover the position of each articulator as accurately as possible at all times. However, it is not guaranteed whether following this path, by incorporating more and more constraints, will lead to a perfect method for performing the inversion mapping. In other words, it may be the case that no number of constraints that may be reasonably formulated and applied will be able to disambiguate completely the movements of articulators recovered from the acoustic signal.

Consider the case where a range of articulatory configurations (or articulatory "fiber") may produce an acoustic vector, but that within this range, one or more of the articulators may be well defined, while others may vary as has been proposed by [1]. This may be demonstrated by the hypothetical example of the production of an /m/ segment. The lips and the velum are critical to the production of this sound. However, the movement of the tongue is not critical to producing an /m/ segment. During the bilabial closure, the tongue could take any number of positions. The exact movements of the tongue will presumably depend to a large extent on the neighbouring sounds (anticipatory articulation, coarticulation and so on). Knowing these, it might be possible to make a best guess at how the tongue moves during the time where it is non-critical, however, there will arguably always be a some degree of uncertainty in this estimate.

The motivation for the approach described in this paper differs significantly from the apparent aim of previous work. In addition to employing methods to disambiguate instantaneous non-uniqueness, it is the intention of this approach to explicitly model the uncertainty, or *variance*, around an estimate of an articulator's position. This approach is 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.

3 HUMAN ARTICULATORY DATA

In order to investigate the acoustic-to-articulatory inversion mapping, researchers have turned to analytical methods and articulatory synthesis models among others. Unfortunately, these approaches are typically afflicted by certain fundamental difficulties. First, there is no obvious way of assessing the accuracy of the inferred vocal tract area function. Second, there is the related difficulty in deciding whether a given estimate of the vocal tract area function is even physiologically possible, let alone likely! Third, there exists the danger that limitations of the analysis or articulatory synthesis models can exert a deleterious effect and impede progress. For example, inverse filtering methods are only really suited to a subset of phone types: vowels and voiced consonants. Major difficulties arise for analysis during nasalised sounds, where the velum lowers and the oral and nasal cavities become coupled. Voiceless sounds are likewise ill-suited to analysis.

Measured human articulatory data is an extremely valuable resource for helping to develop an inversion mapping method. It is arguably much more useful than data generated by articulatory synthesis models. Synthesised data may contain artifacts resulting from limitations and inaccuracies in the generating model. What is more, we are not left with the same difficulties in assessing how an inversion method is *really* performing. There is no more accurate production model for a speech signal that the vocal tract that actually produced it! We can assess the performance of an inversion algorithm by comparing the output with how the speaker actually articulated an utterance.

Despite the obvious advantages, surprisingly limited work has been done on the inversion mapping using measured human articulatory data. The studies that have been done have focused almost entirely on a restricted set of speech sounds only. There is fortunately little overlap between these restricted sets, and therefore we are taken some way to hoping that inversion is possible for all speech sounds. However, there is no substitute for actually attempting inversion for all speech sounds at once and for continuous speech. Attempting this and reporting the results is a very necessary research step.

3.1 MOCHA

The **M**ulti**ch**annel **A**rticulatory (MOCHA) database is currently being recorded in the sound damped studio at the Edinburgh Speech Production Facility based in the department of Speech and Language Sciences, Queen Margaret University College, UK ([18]). By May 2001, the MOCHA database is intended to feature forty speakers with a variety of regional accents. So far, two speakers have been made available, one male (with a Northern English accent) and one female speaker (with a Southern English accent).

While the subject speaks, four data streams are recorded concurrently straight to computer: the acoustic waveform (16kHz sample rate, with 16 bit precision) together with laryngograph, electropalatograph and electromagnetic articulograph data. The electromagnetic articulograph samples the movement of receiver coils attached to the articulators in the midsagittal plane at 500Hz. Coils are attached to the top lip, bottom lip, bottom incisor, tongue tip, tongue body, tongue dorsum and velum. Additional coils are attached to the bridge of the nose and the upper incisor, but the signals from these coils, which should have minimal movement relative to each other, are only used as part of an algorithm which processes the other EMA channels to correct for head movement.

The speaker is recorded reading a set of 460 British TIMIT sentences. These short sentences are designed to provide "phonetically-diverse" material to maximise the usefulness of the data for speech technology and speech science research purposes. It is intended to capture with good coverage the main connected speech processes in English, for example assimilation.

For this paper, the acoustic waveform and EMA data recorded for the female speaker (`fsew0`) has been used.

3.2 Processing

In order to render the raw parallel articulatory and acoustic data into a format suitable for use with neural networks, several processing steps were carried out. First, filterbank analysis was carried out on the acoustic signal, using a Hamming window of 20ms with a shift of 10ms. For each time frame, the acoustic vector consisted of 20 mel-scale filterbank coefficients. These were normalised across all 460 utterances to lie within the range [0.0, 1.0]. The EMA traces were downsampled to match the 10ms shift rate of these acoustic feature vectors. This was done by first lowpass filtering the signal (forwards, then backwards to counteract phase distortion). The articulatory feature vectors were normalised to lie in the range [0.1, 0.9] for training the MLP. This is because the logistic activation function of the output units in this network has the unrealisable asymptotic limits of 0.0 and 1.0. For the mixture density network, which we shall see has a linear activation function for the equivalent output units, the articulatory feature vectors were normalised to lie in the range [-1.0, 1.0]. From the 460 utterances contained in the database of speaker `fsew0`, 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 training set contained 92,557 pairs of acoustic and articulatory feature vectors.

4 INVERSION BY MLP

A neural network, once trained, requires only modest computational resources relative to other models in terms of both memory space and speed of execution. This factor has spurred interest in neural networks for several researchers working on the acoustic-to-articulatory inversion mapping. For example, [15] cite this as a major motivation for working with neural networks. The total memory requirement of their trained assembly of neural networks was just 4% of that required by the codebook they used for comparison, without any perceived loss of quality. In addition, the network system

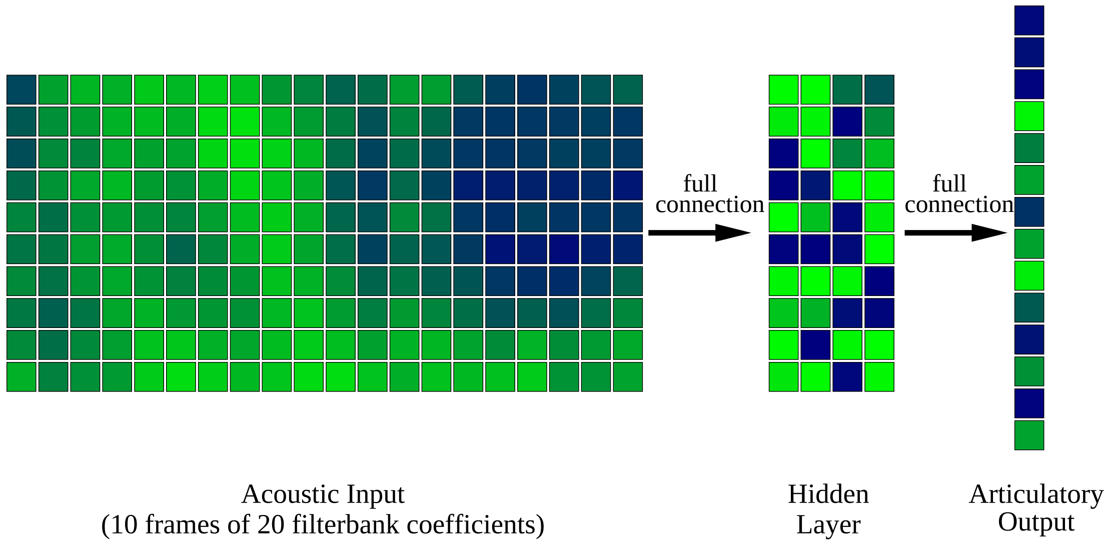

Figure 1: Feedforward MLP for performing the inversion mapping. This network shows the presentation of an acoustic input vector, made up of 10 time frames of 20 filterbank coefficients. Each frame of filterbank coefficients was computed with a shift of 10ms, therefore the total context window of acoustic input applied to the network is approximately 100ms.

provided a mapping from acoustics to articulatory parameters 20 times faster than the codebook lookup.

The low computational cost of neural networks relative to other methods is still a highly desirable property, even in this day of ever more powerful computers. The work described by [14], [19], [15] and [9] in particular, has provided very useful insight into how we might expect neural networks to perform the acoustic-to-articulatory inversion mapping, and that there is a strong case for attempting a neural network mapping on greater quantities of phonetically diverse speech.

4.1 Architecture

The topology of the feedforward MLP used in this paper is shown in Figure 1. As discussed in Section 2, the instantaneous mapping from acoustics to articulation is liable to contain multiple solutions. The use of a context window in the acoustic input domain is intended to alleviate this instantaneous non-uniqueness.

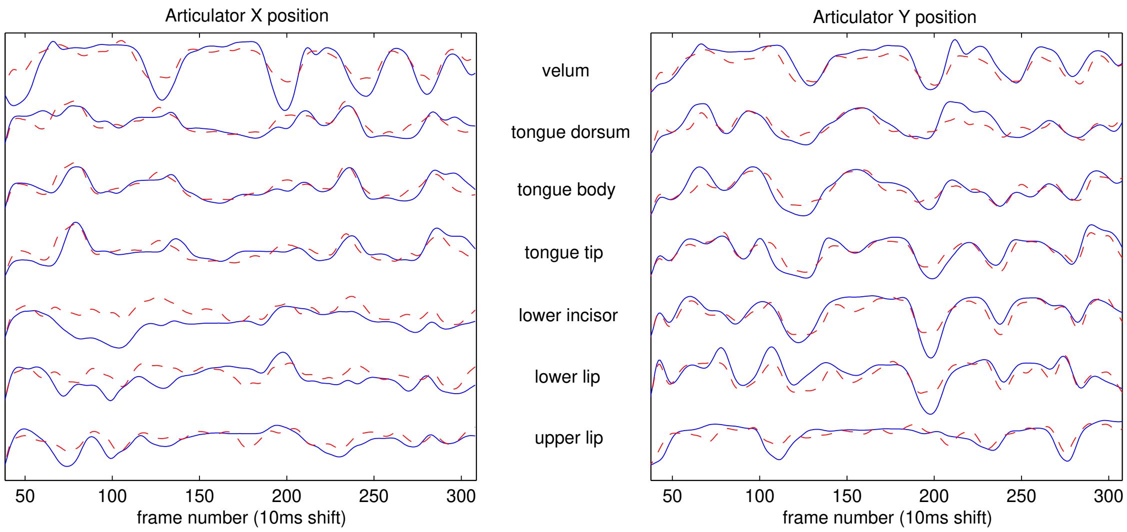

Figure 2: Comparison of real and estimated articulatory trajectories. The test utterance shown here is “Masquerade parties tax ones imagination”. The real articulatory trajectory is shown as the solid line, while the trajectory estimated by the MLP is shown as the dashed line. The silence at the beginning and end of the utterance within the file have been removed.

4.2 Training

The MLP shown in Figure 1 was trained using the Scaled Conjugate Gradients ([12]) algorithm until the error ceased to reduce on the separate validation data set. 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. As with standard backpropagation, conjugate gradient methods iteratively approach a minimum in the error function. However, whereas standard backpropagation always proceeds in the direction of the gradient of the error function, a conjugate gradient method will proceed in a direction which is conjugate to the directions of the previous steps. Thus, the minimisation performed at one step is not partially undone by the next. It is the generally held view that second order techniques find a better way to a (local) minimum than first order techniques, although they incur higher computational cost at each cycle of training. Scaled Conjugate Gradients has been shown to be considerably faster than standard backpropagation and other conjugate gradient methods ([12]).

Articulator	RMSE (mm)
upper lip x	0.8
upper lip y	1.0
lower lip x	1.0
lower lip y	2.2
lower incisor x	0.7
lower incisor y	1.0
tongue tip x	2.1
tongue tip y	2.2
tongue body x	1.9
tongue body y	1.9
tongue dorsum x	1.8
tongue dorsum y	2.1
velum x	0.4
velum y	0.4

Table 1: Root mean square error (in millimetres) between real and network estimated trajectories for the unseen test set.

4.3 Results

Figure 2 shows a comparison between the real and MLP estimated EMA trajectories for an unseen sentence from the test set. The sentence reads "Masquerade parties can tax ones imagination." It is important to note that these trajectories (both the estimated and real EMA in fact) have been low-pass filtered at 15Hz. Smoothing in this way can be viewed as applying an articulatory constraint to the sequence of articulatory configurations recovered by the network at each time frame ([7]).

As evident in this example, 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.

Table 1 gives a more quantitative impression of how the MLP performs the acoustic-to-articulatory inversion mapping. The average of the RMSE values shown is 1.4 millimetres. These results compare favourably with previous reports comparing estimated articulatory trajectories with measured, human trajectories; [7] report an error around 2mm for the tongue points; [5] reports average RMS error of around 2mm; [13] report RMS error between estimated and actual articulatory trajectories of about 1.65mm on average.

5 MIXTURE DENSITY NETWORKS

We saw in Section 4 that the MLP performed on a par with other inversion methods reported in the literature. However, it is very apparent that the network was better in some places than in others, and better for some articulators than for others.

It is well understood that the output of an MLP trained by minimising the sum-of-squares error function approximates the conditional average of the target values in the training data ([3]). This is problematic in two respects. First, what if the target data were to have a bimodal distribution? The average of those points according to the least squared error solution may not actually be close to either cluster! The MLP does not have the power to model distributions of target points any more complex than a unimodal Gaussian. Second, the MLP only provides the mean value. We do not receive any indication as to the variance around that mean. For example, the MLP is not able to distinguish between the case where the target values corresponding to a given input are clustered tightly round their mean and the case where they are spread throughout a large region about the mean. In short, there is unfortunately no way of knowing when the MLP output is likely to be accurate and when to believe it less.

In this section, we describe an approach which allows us to capture both the multimodal aspects of the inversion mapping, as well as give the variance around the estimated articulatory positions. Since mixture density networks 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 [3], [2].

5.1 Theory

A mixture density network is obtained by combining a conventional neural network with a mixture density model. An example MDN is shown in Figure 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 3 components, so that it is given by:

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

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 jth kernel, and $\alpha_j(\mathbf{x})$ is the mixing coefficient for the jth kernel. The mixing coefficients can be thought of as the *prior* probability that a target vector $\mathbf{t}$ has been generated by the jth kernel. 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 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 9 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. Notice that here we are using Gaussian components with spherical covariance (hence only 1 variance pa-

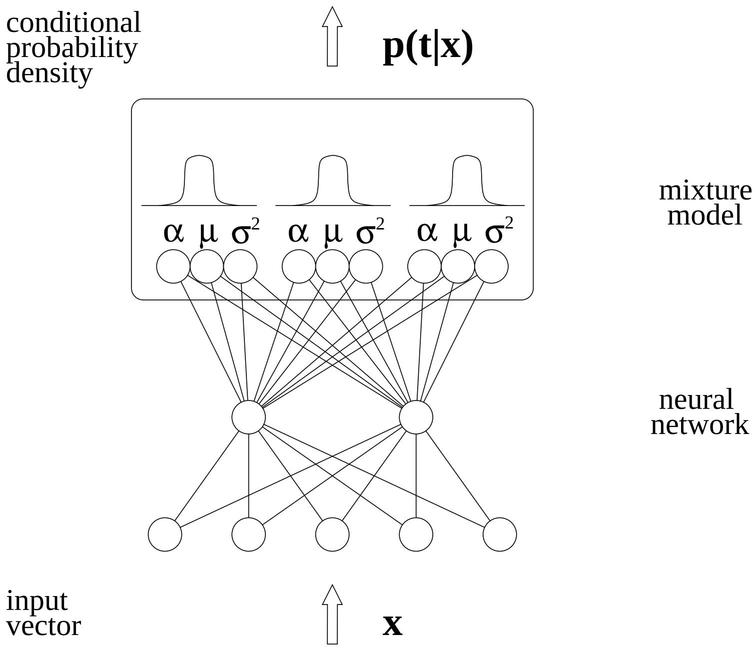

Figure 3: The mixture density network is a combination of a mixture model and a neural network. In a trained MDN, the neural network maps from the input vector $\mathbf{x}$ to the control parameters of the mixture model, which in this case uses Gaussian components (priors α, means μ and variances σ^2)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

rameter 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 Gaussians is theoretically able to model any distribution function with arbitrary accuracy assuming enough components are available ([2]).

Clearly, the mixing coefficients must lie in the range $0 \leq \alpha_j(\mathbf{x}) \leq 1$ and sum to one in order to represent probabilities. This is achieved by using the *softmax* function to relate the mixing coefficients of the mixture model to the output of the corresponding units in the neural network:

$$\alpha_j = \frac{\exp(z_j^{\alpha})}{\sum_{l=1}^{M} \exp(z_j^{\alpha})} \tag{2}$$

where z_j^{α} is the output of the neural network corresponding to the mixture coefficient for the jth mixture component. The variances of the mixture model are related to the corresponding outputs of the neural network according to the following function:

$$\sigma_j = \exp(z_j^{\sigma}) \tag{3}$$

where z_j^{σ} is the output of the neural network corresponding to the variance for the jth mixture component. Finally, the means parameters for the mixture model are represented directly by the corresponding outputs of the neural network:

$$\mu_{jk} = z_{jk}^{\mu} \tag{4}$$

where z_{jk}^{μ} is the value of the output unit corresponding to the kth dimension of the mean vector for the jth mixture component.

The object of training the MDN will be to minimise the negative log likelihood of the observed target data points given the mixture model parameters:

$$E = -\sum_n \ln \left\{ \sum_{j=1}^{M} \alpha_j(\mathbf{x}^n) \phi_j(\mathbf{t}^n | \mathbf{x}^n) \right\} \tag{5}$$

Since it is the neural network which 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 (given in [2]). 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 non-linear optimisation problem to which standard non-linear optimisation algorithms can be applied.

In essence, the mixture density network gives us a principled method for modelling a full conditional probability density of the target data for each input vector.

5.2 Architecture

Like the MLP previously seen, the example MDN used in this paper uses a context window of approximately 100ms, which means it has 200 input units. The hidden layer contains 40 units, with sigmoid activation function. In the target articulatory domain, 16 mixture components with spherical covariance are used. These cover the 14 dimensions of the articulatory domain.

5.3 Training

The Mixture density network was trained on the same parallel acoustic-articulatory data as the previous MLP. The error function in Equation 5 was minimised using the Scaled Conjugate Gradients non-linear optimisation algorithm until the error on a separate validation set was minimised. Prior to training, the network was first initialised by a k-means based initialisation algorithm. For this initialisation algorithm, the weights for the network are first randomised by sampling from a Gaussian. Then, a Gaussian mixture model of the same form as the MDN output is used to model the **unconditional** density of the target data. The k-means algorithm is used to determine the component centres. The priors are computed from the proportion of the target data belonging to each component, and the variances are calculated as the sample variance of the target data points belonging to each component from the associated mean. Finally, the network biases are adjusted so that the net will output the values in the Gaussian mixture model. For the example MDN presented here, 10 iterations of the k-means algorithm were used.

5.4 Results

Figure 4 gives a demonstration of the output of the MDN described above. Figure 5 shows the probability density over the range of tongue dorsum y movement for one frame taken from the second [s] in the utterance. Note that the MDN estimates that the value for the tongue dorsum height lies within a fairly constrained range. Compare this distribution with that shown in Figure 6. This wider distribution suggests the MDN finds it harder to estimate what exactly the tongue dorsum height will be during the production of an [m] phone. This is not unreasonable; the back of the tongue is not critical to the production of an [m], and it may vary without having a profound effect on the acoustic signal produced.

Looking at Figure 4, it should be apparent that where the variance is low (i.e. during dark regions), the accuracy of the inferred height of the tongue dorsum is fairly good. On the other hand, during

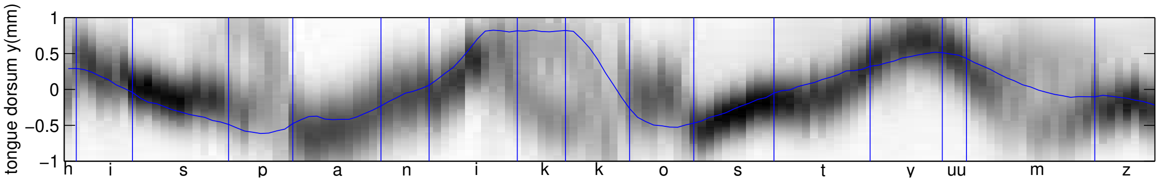

Figure 4: A “probabilitygram” of part of the unseen test utterance “Hispanic costumes are quite colourful.” This plot shows the probability density for the height of the tongue dorsum over its range of movement as a function of time. Intense black indicates high probability density. The real, measured trajectory of the tongue dorsum y parameter is overlaid for comparison.

sequences of framewise distributions where the centre of probability mass seems to err away from the correct trajectory, we see larger variance (i.e. during lighter regions). This perspective is more easily appreciated in Figure 7, where a plot of framewise variance is juxtaposed with framewise square error. The error is calculated as the square of the distance between the real trajectory and the conditional average trajectory. The conditional average trajectory may be computed from the Gaussian mixture distribution at each time step as follows:

$$\sum_j \alpha_j(\mathbf{x})\mu_j(\mathbf{x}) \tag{6}$$

where $\alpha_j(\mathbf{x})$ is the prior, or mixture coefficient, for the jth mixture component, and $\mu_j(\mathbf{x})$ is the mean for the jth mixture component. Meanwhile, the variance of the density function about the conditional average at each time step is given by:

$$\sum_j \alpha_j(\mathbf{x}) \left\{ \sigma_j(\mathbf{x})^2 + \left\| \mu_j(\mathbf{x}) - \sum_l \alpha_l(\mathbf{x})\mu_l(\mathbf{x}) \right\|^2 \right\} \tag{7}$$

The key point to note in Figure 7 is that there are roughly four sections of the utterance where the conditional average value of the MDN output differs noticeably from the real articulatory trajectory. However, these four sections are all accompanied by elevated variance, which can be interpreted as a lack of confidence in the estimated conditional average position. There are also additional sections of the utterance which exhibit a higher variance. However, during these sections, the conditional average estimate given by the MDN may coincidentally be accurate.

Finally, Figure 8 demostrates that bimodal distributions are indeed present in the acoustic-to-articulatory inversion mapping, and moreover that the MDN is able to model them.

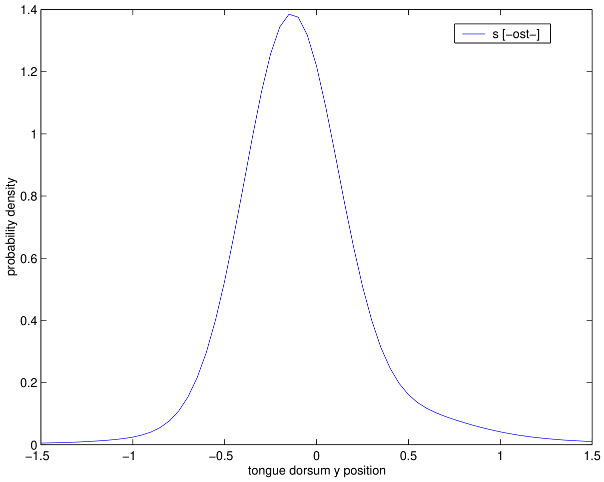

Figure 5: Probability density as a function of tongue dorsum height for one time frame extracted from the [s] in the word [kostyumz]. Compared with the distribution shown in Fig. 6, this distribution has a narrow width, indicating a higher degree of certainty that the tongue dorsum is located around the centre of this distribution.

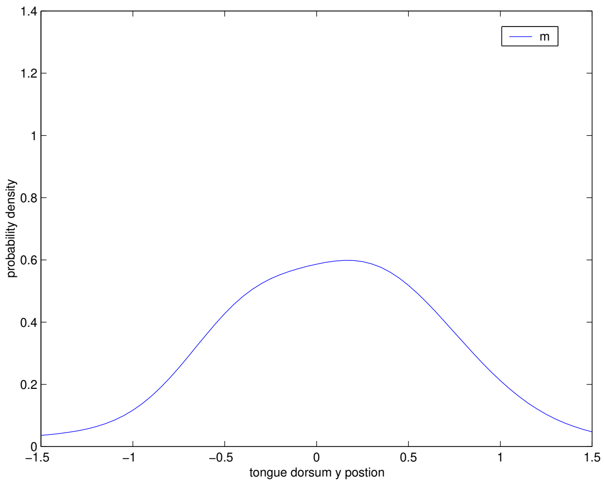

Figure 6: Probability density as a function of tongue dorsum height for one time frame extracted from the [m]. Compared with the distribution shown in Fig. 5, this distribution has a wide width, indicating a lower degree of certainty that the tongue dorsum is located around the centre of this distribution.

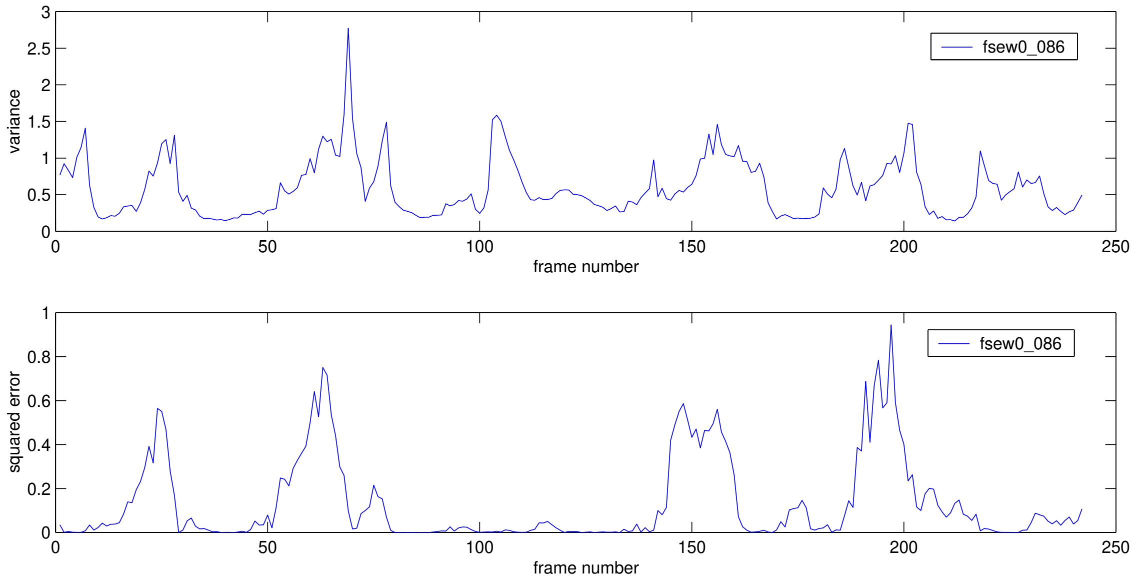

Figure 7: A comparison of accuracy and variance for the unseen test utterance “Hispanic costumes are quite colourfull”. The bottom plot shows the square error between the trajectory of the conditional average tongue dorsum y position calculated from the output of the MDN (see Equation 6) and the real trajectory of the tongue dorsum. The top plot shows the overall variance around the conditional mean (see Equation 7).

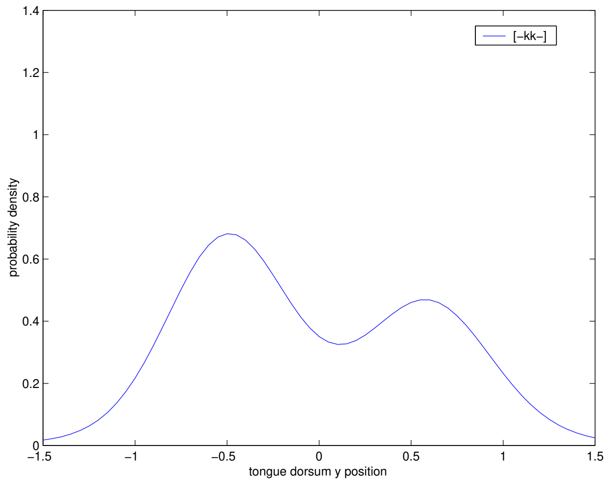

Figure 8: Probability density as a function of tongue dorsum height for one time frame extracted from the [kk] section. Note the bimodal distribution of the tongue dorsum for this phone.

6 DISCUSSION

The MDN gives a whole description of the conditional probability density of the target domain in the form of a mixture model. This density function may be used to compute several different "answers". For example, we can 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 we saw in Section 5.4, we can also compute the variance around this conditional average, which goes beyond the capabilities of a standard MLP. On the other hand, at each time frame, we could take the mean and variance of the Gaussian with the highest prior. This approximates the mixture of experts model ([8]) as another special case.

Hence, MDNs in some sense occupy an ideal middle ground between supervised and unsupervised empirical learning models. They are trained in a way typical of supervised methods. However, at test time, there are similarities with unsupervised techniques. In other words, the required "answer" is postponed until testing time, and as such the MDN offers a powerful modelling advantage over other supervised learning models.

7 CONCLUSIONS

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 real, parallel articulatory-acoustic data.

While the performance of this network compared favourably with the results of other inversion methods reported in the field, there are at least two drawbacks. First, the MLP is limited to modelling target data points roughly approximating a unimodal Gaussian, which is not 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.

To address these issues, we have looked at an example of a feedforward mixture density network. This preliminary investigation has shown that the mixture density network is very well suited to delivering the required functionality for performing the inversion mapping. For example, instances of bimodality in the target domain have indeed been observed. Also, larger variances have been associated with sections of increased error for inferred articulator positions, which we might choose to interpret as a confidence measure.

8 FUTURE WORK

A recurrent extension to the mixture density network has been described: the bidirectional recurrent mixture density network ([17]). Previous experience with Elman-style recurrent networks has shown that the trajectories recovered tend to be smoother and more consistent than those produced by an equivalent MLP. Therefore, it is envisaged that applying recurrent mixture density networks to the inversion mapping may prove fruitful.

Future work will also focus on exploring how best to use mixture density network output for use as a frontend for the articulatory based speech recogniser which is currently being developed at CSTR ([6]).

References

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

[2] Chris Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] Chris M. Bishop. Mixture density networks. Technical Report NCRG/4288, Neural Computing Research Group, Department of Computer Science, Aston University, Birmingham, B4 7ET, U.K., February 1994.

[4] S. Chennoukh, D. Sinder, G. Richard, and J. Flanagan. Voice mimic system using an articulatory codebook for estimation of vocal tract shape. In *Proc. Eurospeech*, pages 429–432, Rhodes, Greece, September 1997.

[5] Sorin Dusan. *Statistical Estimation of Articulatory Trajectories from the Speech Signal using Dynamical and Phonological Constraints*. PhD thesis, Dept. of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, April 2000.

[6] J. Frankel and S. King. Speech recognition in the articulatory domain: Investigating an alternative to acoustic hmms. In *Proc. Workshop on Innovations in Speech Processing (This Volume)*, April 2001.

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

[8] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[9] T. Kobayashi, M. Yagyu, and K. Shirai. Application of neural networks to articulatory motion estimation. In *Proc ICASSP*, pages 489–492, 1991.

[10] R. Kuc, F. Tutuer, and J. R. Vaisnys. Determining vocal tract shape by applying dynamic constraints. In *Proc. ICASSP*, pages 1101–1104, Tampa, Florida, 1985.

[11] B. Lindblom, J. Lubker, and T. Gay. Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation. *J. Phon.*, 7:146–161, 1979.

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

[13] Takesi Okadome, Shin Suzuki, and Masaaki Honda. Recovery of articulatory movements from acoustics with phonemic information. In *Proc. 5th Seminar on Speech Production*, pages 229–232, Kloster Seeon, Bavaria, May 2000.

[14] G. Papcun, J. Hochberg, T. R. Thomas, F. Laroche, J. Zachs, and S. Levy. 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, August 1992.

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

[16] S. Roweis. *Data Driven Production Models for Speech Processing*. PhD thesis, California Institute of Technology, Pasadena, California, 1999.

[17] Michael Schuster. *On Supervised Learning from Sequential Data with Applications for Speech Recognition*. PhD thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, February 1999.

[18] Alan A. Wrench and William J. Hardcastle. A multichannel articulatory speech database and its application for automatic speech recognition. In *Proc. 5th Seminar on Speech Production*, pages 305–308, Kloster Seeon, Bavaria, May 2000.

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