SPEECH RECOGNITION IN THE ARTICULATORY DOMAIN: INVESTIGATING AN ALTERNATIVE TO ACOUSTIC HMMS

Joe Frankel Centre for Speech Technology Research, University of Edinburgh.
Simon King Centre for Speech Technology Research, University of Edinburgh.

1 INTRODUCTION

1.1 Why look at alternatives to the hidden Markov model?

The hidden Markov model (HMM) has proven to be the model which made large-vocabulary automatic speech recognition (ASR) possible. The HMM is robust, versatile and has at its disposal a host of efficient algorithms to deal with training, speaker adaptation and recognition. However there is nothing uniquely speech orientated about the HMM. In fact, certain assumptions are made of speech which are known to be untrue. For example, speech is modelled as a piecewise stationary process when we know it to be continuous. Also co-articulation, which should be a rich source of information, simply provides unwanted variation. This variation is generally taken into account by modelling every phone in every context which in turn leads to problems of data sparcity, making elaborate parameter tying schemes necessary.

1.2 Better modelling through speech production knowledge

Speech modelling generally occurs in the acoustic domain, which is natural given that this is the data we have most ready access to. Any practical speech recogniser must of course take acoustic waveforms as input, however to take these in isolation from the production mechanism which created them ignores a rich source of prior knowledge.

We propose that modelling speech in the articulatory domain would address some of these issues. The data here consists of trajectories which evolve smoothly over time, namely coordinates of points on the articulators. Effects such as co-articulation and assimilation are produced in the articulatory domain and therefore can be modelled explicitly. This is in contrast to looking at these effects in the acoustic domain where they are confounded with the representation.

To model these trajectories we use linear dynamic models (see section 3 for a full introduction). These are particularly well suited to modelling smoothly varying, continuous yet noisy trajectories. We have access to real articulatory data, collected by Alan Wrench at Queen Margaret college, Edinburgh (see [8] for further details). This has been used to train neural networks to recover articulatory traces from the acoustics. In our experiments we have used both real and automatically recovered articulation.

2 DATA

The data consists of a corpus of 460 TIMIT sentences for which parallel acoustic-articulatory information was recorded using a Carstens Electromagnetic Articulograph (EMA) system. Sensors were placed at three points on the tongue (tip, body and dorsum), upper and lower lip, jaw and also the velum. Their position in the midsagittal plane was recorded 500 times per second and the acoustic signal sampled with 16 bit precision at 16 kHz. 30% of the sentences were set aside for testing and 70% used for training. The data was labelled using an HMM based system where flat-start monophone models were forced-aligned to the acoustic data from a phone sequence generated by a keyword dictionary [8].

2.1 Automatic estimation of articulatory parameter values

Other work at CSTR has used neural networks to perform the acoustic to articulatory inversion mapping. The simulated articulatory traces used in these experiments were generated using a recurrent neural network with a 200ms input context window and 2 hidden layers. A single output unit was used for each articulator coordinate (ie one for x, one for y), and the networks were trained on simultaneous streams of acoustic and articulatory data. For details see [3]

2.2 Feature set

A	EMA
B	EMA + zero crossings + voicing
C	EMA + 12 cepstra + energy
D	EMA + 12 cepstra + energy + zero crossings + voicing
E	12 cepstra + energy

Table 1: summary of the different feature sets used for experimentation.

Using a feature set consisting only of articulatory parameters lacks certain information. For instance making a voiced/voiceless classification, or indeed spotting silences, is compromised by the lack of voicing information and energy. We have experimented with augmenting the feature set to use other parameters: mel-scale cepstral coefficients, energy, (acoustic waveform) zero crossing rate, and a voiced/voiceless classification. High values for the zero crossing rate signify noise, i.e. frication and low values are found in periodic, ie voiced sections of speech. The voiced/voiceless decision was made from a laryngograph trace using a pitchmarking tool. The different experimental configurations are given in Table 1. Both real and simulated EMA traces were used, and feature set E, just the cepstral coefficients, was included for comparison.

3 LINEAR DYNAMIC MODELS

As mentioned in section 1, we have chosen a linear dynamic model to model the articulatory trajectories. This is a model which appears in a whole host of applications with a variety of names (Kalman model, Dynamic system model, Continuous state linear Gaussian system etc). It is described by the following pair of equations:

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \mathbf{w}_t \qquad (1)$$

$$\mathbf{y}_t = H\mathbf{x}_t + \mathbf{v}_t \qquad (2)$$

with $\mathbf{w}_t \sim N(\mu_w, Q_w)$ and $\mathbf{v}_t \sim N(\mu_v, Q_v)$

The basic premise of the model is that there is some underlying dynamic process which can be modelled by Equation 1. This equation describes how $\mathbf{x_t}$, the state variable at time t, evolves from one time frame to the next. A linear transformation via the matrix F and the addition of some Gaussian noise, $\mathbf{w_t}$, provide this, the dynamic portion of the model.

The complexity of the motion that Equation 1 can model is determined by the dimensionality of the state variable. For example, a 1 dimensional state space would allow exponential growth or decay with an overall drift ($\mu_\mathbf{w}$ can be non-zero) and 2 dimensions could describe damped oscillation with a drift. Increasing the dimensionality beyond 4 or 5 degrees of freedom allows fairly complex trajectories to be modelled.

The observation vectors, given at time t by $\mathbf{y_t}$, represent realisations of this unseen dynamical process. A linear transformation with the matrix H and the addition of measurement noise, $\mathbf{v_t} \sim N(\mu_\mathbf{v}, Q_v)$ (Equation 2) relate the two. The trajectories could be modelled directly. However using a hidden state space in this way makes a distinction between the production mechanism at work and the parameterisation chosen to represent it. In the case of the articulatory data we are working with, fewer degrees of freedom are needed for modelling purposes than are originally present in the data. This is no surprise; for example there are three coils giving us x and y coordinates over time for the motion of the tongue. These six data streams are clearly going to be highly correlated and so there will be redundancy of information.

The models are segment-specific, with one set of parameters H, F, Q_v, Q_w, $\mu_\mathbf{v}, \mu_\mathbf{w}$, and $\mathbf{x_0}$ describing the articulatory motion for one unit of speech, although it is possible to share parameters between models. For practical reasons, the segments used so far have been phones.

The model can be thought of as a continuous state HMM [4]. Having a state which evolves in a continuous fashion, both within and between segments, makes it an appropriate choice to describe speech. Attempts to directly model speech in the *acoustic* domain using LDMs have been made, however the defining feature of these models is that they are able to model smoothly varying (but noisy) trajectories. This makes them ideally suited to describing articulatory parameters. Furthermore, the asynchrony between the motion of different articulators is absorbed into the system, and the critical versus non-critical nature of articulators (see below) is captured in the state to observation mapping covariance Q_v. Lastly, parameter estimation is made much simpler through having a linear mapping between state and observation spaces, which is a reasonable assumption for observations in the articulatory domain.

3.1 Training

The Expectation Maximisation (EM) algorithm is used to train the models. As mentioned in Section 2, we have a time-aligned phonetic transcription of the data. This was used to extract all the training tokens for each segment type. EM was then used to train the parameters of an LDM on each of these subsets of the data. Each iteration consists of two stages. Firstly in the E-step statistics are accumulated over the training tokens, using the most recently estimated parameter values. This is followed by the M-step in which these statistics are used to update the model parameters. See [4] for mathematical details.

Overfitting occurred fairly rapidly; models trained on simulated articulatory parameters in general needed 5-7 iterations of the EM to converge, whereas 3-4 was sufficient for models trained on real data.

3.2 Critical versus non-critical articulators

The behaviour of the lips and velum have a fundamental role in the production of a /p/, whereas the motion of the tongue is far less important. Here the lips and velum are *critical* articulators and the tongue *non-critical*. It was reported in [5] that ’critical articulators are less variable in their movements than non-critical articulators’.

Examination of model parameters shows evidence of this effect. For example, Figure 1 shows variance terms for selected articulatory streams from trained models of the three voiced oral stops in English. These variances are the noise terms associated with the transformation from the hidden space to the observation space, and indicate the confidence the model places on its prediction of the articulator’s position. The more consistent (critical) an articulator, the more confidence the model will have in predicting its position, and this will be expressed with a lower variance term.

If we first consider the velum, which is critical for all three voiced stops, we see that the variance of its movement is uniformly low for all three models. However, the picture is different for unshared critical articulators: for the /b/ model, the lower lip has the lowest variance; for the /d/ and /g/ models, it is the tongue tip and tongue dorsum respectively that show the lowest variance.

These findings tie in with the notion of critical articulators, and also offer insight into the nature of the acoustic-to-articulatory mapping necessary for a speech recognition system. Lower significance will be placed on the position of a non-critical articulator and so more emphasis should be put onto faithfully recovering important features of a segment rather than recovering all articulation perfectly

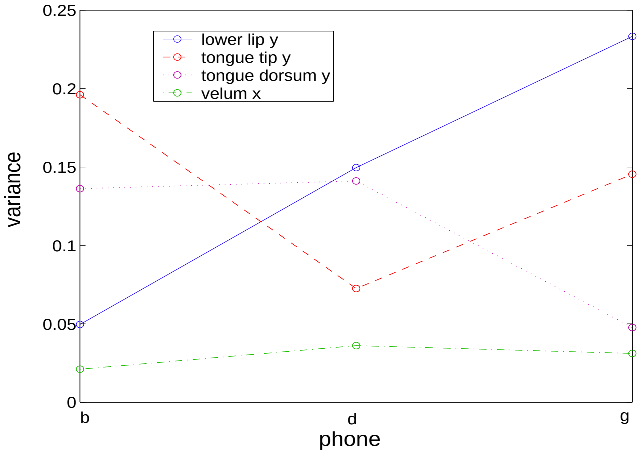

Figure 1: Variances put on the projection from state to observation space for normalised, real EMA data on segments /b/, /d/, and /g/

all of the time.

It is worth pointing out that non-critical articulators are as useful as critical ones in characterising and distinguishing segments as the model learns to put different emphasis on different parts of the data stream.

4 CLASSIFICATION

To make a decision as to which model generated each segment of the unseen data, we need to calculate a likelihood for each of the competing models. This is the likelihood of the observations given the model parameters and cannot be computed directly as the LDM has a hidden state space. However an approximation to this likelihood can be computed if we first infer values for these ’missing’ parameters. We have experimented with two approaches to this task. The first was by using the expected state values computed in the forward Kalman recursions, as in the E-step of the EM algorithm. The other, which we found to be marginally more successful, is to use the posterior predictive distribution of the state variable, $\mathbf{x_t}|\mathbf{Y_t}$, where $\mathbf{Y_t} = (\mathbf{y_1}, \ldots, \mathbf{y_t})^T$.

Once computed, a Viterbi search using a bigram language model chooses the most likely model.

4.1 Results

There was some degree of flexibility in the dimensionality chosen for the state space. Figure 2 shows how raw (no language model) classification scores are affected by varying the state dimension. The

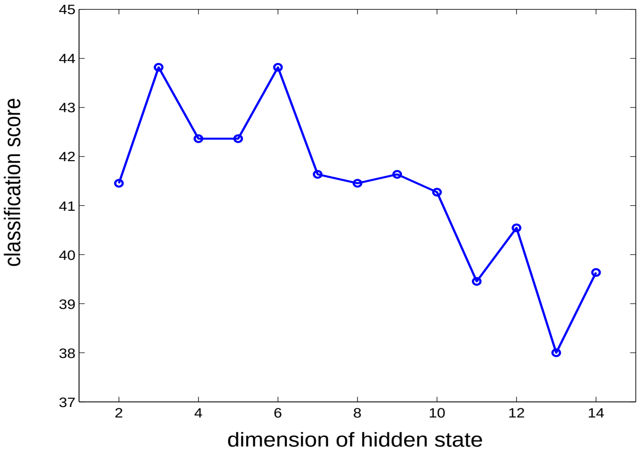

Figure 2: Raw (no language model) classification score against state dimension for a validation set consisting of 20 utterances. Models were trained and tested on real EMA data only.

feature set in use here is the real EMA data. Between 3 and 9 degrees of freedom seems ideal, with higher dimensionalities and correspondingly higher numbers of parameters producing slightly worse results.

data	feature set	accuracy
real articulatory	A	51%
	B	63%
	C	77%
	D	74%
simulated articulatory	A	46%
	B	47%
	C	55%
	C*	64%
	D	56%
acoustic	E	68%

Table 2: Classification results for a 46 phone model set using both real and simulated articulatory data. * denotes trained on real, tested on simulated data

Table 2 summarises the results of experimentation with the system. For each system, the number of training iterations and the dimension of the hidden state was optimised. The best result for each is the one quoted.

Training and testing models on the real articulatory data produced a classification score of 51%. Augmenting the feature set to also include zero crossing rate and the voiced/unvoiced decision gave a 12% improvement with a result of 63%. The best result, 77% came from feature set C, which consists of the articulatory and acoustic data. It seems slightly anomalous then that including the zero-crossing rate and voicing decision should reduce this score to 74%. However this is a marginal drop in performance.

Replicating these experiments using the automatically estimated articulatory parameters gave a slight drop in performance. 46% and 47% were the scores based on using the articulatory data only and then including the zero crossing rate and voicing decision. Adding the cepstra to the automatically produced articulation gave a score of 55%, and 56% was obtained for feature set D, which also includes zero crossing rate and the voicing decision. Using the acoustic observations only as input (E) gave a result of 68%.

4.2 Discussion

The first thing to note is that adding real articulatory information to the acoustic data gave a 9% (7.5% relative) increase in classification performance. This supports the notion that articulatory information has the potential to be valuable to ASR. Other studies have also found evidence of this, for example see [7].

From the scores of the systems which use only articulation as input, we see that the real articulatory data performs better than the simulated, although the difference is not huge. However, the performance of the acoustic system alone is better than that of the system which also includes recovered articulatory information. We believe the reason that the data becomes more confusable in combination is that the neural network mapping provides an *average* articulatory configuration given a sequence of acoustic observations.

This means that for a given segment type, a non-critical articulator which in the training set can follow many different trajectories is in fact given a consistent set of paths in the test set (network output). We have been investigating techniques for producing multi-modal output from the networks to deal with this.

The models trained on real data give a better score than the models trained on the recovered data, when both perform classification on recovered traces (see C* in the table 2). Furthermore, the models trained on simulated articulation show lower variance terms than their real data trained counterparts. These findings support the notion that the networks learn consistent patterns when there should be none, i.e. making all articulators ’critical’ all of the time.

5 PHONE RECOGNITION

Recent work has been to use the models for recognition, rather than just classification. For this task we have implemented a stack decoder which is very similar to that in [6].

5.1 Stack decoding for linear dynamic models

The search algorithm is built around a tree-structured lexicon which means that computation can be shared by paths which have common prefixes. For example the words /bit/ and /bik/ would share computation of likelihoods for the phone sequence /b/ /i/.

The stack consists of an ordered heap which holds a number of partial phone hypotheses. These hypotheses each contain a phone sequence, a likelihood for this sequence, and an estimate of the remaining likelihood to the end of the utterance. Clearly the longer the hypotheses, the lower its likelihood will be, so by computing the sum of the two likelihoods (one computed for the phone sequence so far and the other an estimate of what remains), it is possible to compare hypotheses of different lengths.

At each cycle of the algorithm the best partial phone hypothesis is 'popped' from the stack, extended by every allowable phone, and these new hypotheses 'pushed' back. Pruning then throws away unlikely paths to keep the heap size down. The time-asynchronous ordering of the search means that a minimum of time is used in exploring unlikely paths.

It was mentioned in Section 3 that the linear dynamic model could be thought of as a continuous analogue to the HMM. However, there is one crucial difference which affects the task of searching for the best phone alignment. To illustrate this it is useful to make the following definition:

A Markov chain is a random process which moves from state to state, where given the present, the past and future are independent. Letting x_t represent the state at time t, this can be written as:

$$P(x_{t+1}|x_t, x_{t-1}, \dots, x_0) \quad = \quad P(x_{t+1}|x_t) \tag{3}$$

Now an HMM is a Markov chain where each state emits samples from a probability distribution over the observations. This makes the assumption that speech is made up of a series of locally invariant regions, represented by the states. The present state affects the probability with which the system moves to each other state, however not the realisation i.e. output of that state.

Property (3) also holds for LDMs, where x_t is now the state vector at time t, and so as with the HMM all state information is encompassed in the present time. However, unlike the HMM which can jump from one state to any other, the LDM is constrained to follow a trajectory which will depend on x_0, its value at the start of the segment. This means that for decoding purposes, a separate Kalman smoother has to be run for each candidate segment start time. In our implementation, the cost of this extra computation is reduced by caching the probabilities for each model and each start time as they are computed.

5.2 Results and discussion

So far we have implemented the decoder and have a preliminary result. On the same task as before, using feature set D, we achieve a phone accuracy of 50%.

data	feature set	correct	accuracy
real articulatory	D	57%	50%

Table 3: Recognition scores based on real articulatory data.

This result should be treated as work in progress, as the implementation of the decoding is not yet complete. The first addition to be made is that of an explicit duration model. The phone classification score for the same models and feature set is considerably better (74%) than that for recognition (50%) which suggests that the segmentation needs to be improved. Duration modelling is implicit in the LDM, as likelihoods peak at the end of regions which have been explained well by the model. For example, you would expect the likelihood for a /b/ model to peak toward the end of a /b/ in the data. Overlaying an explicit duration model would emphasise these peaks, improving the model's power to choose segment boundaries. This addition to the segment model is straightforward and is assumed integral in [1].

Furthermore, at present the state space is continuous within, but not between segments. x_t is reset (to a value learnt during training) at the beginning of each segment. It should in fact be initialised to the last state value of the phone it is following. In the future each partial hypothesis in the stack will include state vectors corresponding to the candidate end times.

The decoder will also be used for Viterbi training. This involves alternately updating model parameters and then re-segmenting according to the most recent models. Full embedded EM training is impractical for the LDM as a separate forward-backward Kalman smoother would be needed for every possible alignment of models. For further details see [2]. It is expected that Viterbi training will improve performance, as to date the models have been trained using alignments from an acoustic HMM system. This is likely to be different to the segmentation produced using LDMs and a combination of articulatory and acoustic gestures.

6 Conclusion

Our classification scores demonstrate that a combination of articulatory and acoustic features gives a better performance than either does singly. This encourages us to explore the use of articulatory modelling for ASR further.

Using a feature set comprising articulatory and acoustic derived observations begs the question of what units the system should be based on. If phones were used, the articulatory and acoustic portions of the feature set would produce slightly different segmentations. As such we intend to investigate alternative units which reflect the nature of the data better. The phenomenon of co-articulation confounds phone-based systems; however a longer unit based on articulatory features would be capable of storing a certain amount of co-articulation information within it.

Our use of the decoder is in its infancy, however shows promise. We anticipate that adding an explicit duration model will improve performance, as will Viterbi training.

A practical speech recognition system cannot in the end rely upon real articulatory data. We are using the data to take advantage of the useful properties it possesses; smoothly changing trajectories, built-in context information etc., but really it can only be seen as a development tool. As the recogniser grows in scale, the articulatory aspect of the system will be reduced to that of a latent variable, and the two parts of the system will be trained together.

References

[1] V Digalakis and M Ostendorf. Fast algorithms for phone classification and recognition using segment-based models. *IEEE Trans. on Speech and Audio Processing*, 40(12):2885–2896, 1992.

[2] V. Digilakis, J. Rohlicek, and M. Ostendorf. ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Trans. Speech and Audio Processing*, 1(4):431–442, October 1993.

[3] J Frankel, K Richmond, S King, and P Taylor. An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces. In *Proc. ICLSP*, 2000.

[4] M Ostendorf, V Digalakis, and O Kimball. From HMMs to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 1996.

[5] 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.

[6] A. Robinson, G. Cook, D. Ellis, E. Fosler-Lussier, S. Renals, and D. Williams. Connectionist speech recognition of broadcast news. *Speech Communication*, forthcoming 2001.

[7] Alan A. Wrench. A new resource for production modelling in speech technology. In *Proc. Workshop on Innovations in Speech Processing*, 2001.

[8] 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.