ASR - Articulatory Speech Recognition

Joe Frankel, Simon King

Centre for Speech Technology Research
The University of Edinburgh
joe@cstr.ed.ac.uk

1. ABSTRACT

We describe a speech recognition system which uses a combination of acoustic and articulatory features as input. Linear dynamic models capture the trajectories which characterise each segment type. We describe classification and recognition tasks for systems based on acoustic data in conjunction with both real and automatically recovered articulatory parameters.

2. INTRODUCTION

The hidden Markov model (HMM) has proven to be the model which has made large-vocabulary automatic speech recognition (ASR) possible. The HMM is robust, versatile and has at its disposal a host of efficient algorithms which 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.

Speech is generally modelled in a parametrised version of 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 using linear dynamic models (see section 4) will 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 coarticulation and assimilation are most simply described in articulatory terms, as opposed to in acoustic terms where they are confounded with the representation. Models that work in the articulatory domain are therefore able to explicitly model these phenomena. We have access to real articulatory data, collected by Alan Wrench at Queen Margaret College, Edinburgh (see [1] 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.

3. 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 mid-sagittal 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 force-aligned to the acoustic data from a phone sequence generated by a keyword dictionary [1].

3.1. Automatic estimation of articulatory parameters

Other work at CSTR has used neural networks to perform the acoustic to articulatory inversion mapping. The automatically estimated 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 (i.e. one for x, one for y), and the networks were trained on simultaneous streams of acoustic and articulatory data. For details see [2].

3.2. Feature set

Using a feature set consisting only of articulatory parameters lacks certain information. For instance, making a voiced/voiceless decision or spotting silences is difficult from what is essentially a silent movie of speech. In order to overcome this, we have experimented with augmenting the articulatory feature set to also include mel-scale cepstral coefficients.

4. LINEAR DYNAMIC MODELS

As mentioned in section 2, we have chosen a linear dynamic model (LDM) to model the articulatory trajectories. The following pair of equations define an LDM:

$$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\epsilon}_t \quad (1)$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t \quad (2)$$

with $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$ and $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$

The basic premise of the model is that there is some underlying dynamic process which can be modelled by Equation 2. 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, $\boldsymbol{\eta}_t$, provide this, the dynamic portion of the model.

The complexity of the motion that Equation 2 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 ($\mathbf{v}$ 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, $\boldsymbol{\epsilon}_t$ (Equation 1) 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 $H, F, C, D, \mathbf{v}, \mathbf{w}$, and $\mathbf{x_0}$ describing the trajectories associated with one unit of speech, although it is possible to share parameters between models. For practical reasons, the segments used so far have been phones, however see section 7 for future intentions.

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 section below) is captured in the state to observation mapping covariance C. 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.

4.1. Training

The Expectation Maximisation (EM) algorithm is used to train the models. The time-aligned phonetic transcription of the data enables us to extract the examples of each segment type from the training set. In the The E-step, statistics are accumulated over these training examples using the most recently estimated model parameters. Then, in the M-step, these statistics are used to update the model parameters. See [3] for mathematical details.

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

4.2. Critical versus non-critical articulators

An articulator which has a fundamental role in the production of a phone is said to be *critical* for that phone. For example, the behaviour of the lips and velum are critical in the production of a [p], whereas the motion of the tongue is far less important.

The variance in the mapping from the hidden to the observation space gives an indication of the confidence the model places on its prediction of an articulator's position. Faith (i.e. low variance) is placed in the prediction of an articulator's position if it is known to behave consistently. It was reported in [4] that 'critical articulators are less variable in their movements than non-critical articulators'. Examination of model parameters shows evidence of this effect. In general, for a given segment type, low variances are assigned to the data-streams corresponding to critical articulators.

This suggests that emphasis should be put onto faithfully recovering important features of a segment in the acoustic-articulatory mapping, rather than recovering all articulation perfectly all the time. Non-critical articulators are however a rich source of information, as the model learns to put different emphasis on different parts of the data stream.

5. CLASSIFICATION

Classification of pre-segmented data can be performed using the maximum a posteriori (MAP) rule, details of which can be found in [5]. This quantity is the likelihood of the observed data given the model parameters. Once a likelihood has been computed for each competing model, a Viterbi search using a bigram language model chooses the most likely phone sequence.

5.1. Results

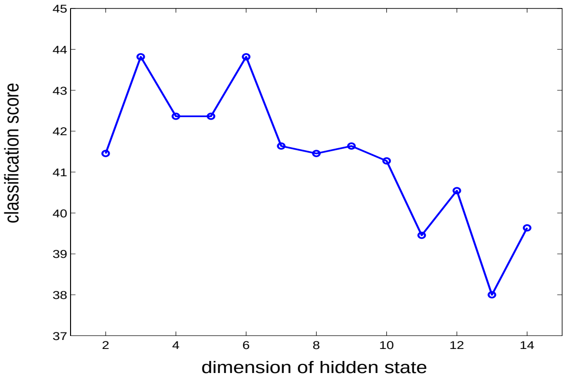

Figure 1: Raw (no language model) classification score against state dimension for a validation set consisting of 20 utterances. Models were trained and tested on real EMA data only.

There was some degree of flexibility in the dimensionality chosen for the state space. Figure 1 shows how raw (no language model) classification scores are affected by varying the state dimension. The feature set in use here is the real EMA data. Between 3 and 9 degrees of freedom seem ideal, with higher dimensionalities and correspondingly higher numbers of parameters producing slightly worse results.

feature set	accuracy
real EMA	51%
real EMA + cepstra + energy	77%
simulated EMA	46%
simulated EMA + cepstra + energy	67%
cepstra + energy	68%

Table 1: Classification results for a 46 phone model set using both real and simulated articulatory data.

Table 1 summarises classification scores for multiple experimental configurations. For each system, the number of training iterations and the dimension of the hidden state was optimised. The best result for each is quoted.

Training and testing models on the real articulatory data produced a classification score of 51%. Augmenting the feature set to include the cepstra and energy, gave the best overall result of 77%.

Replicating these experiments using the automatically estimated articulatory parameters gave a slight drop in performance. The score based on the system using estimated articulation in isolation was 46%, and combining this with the acoustics gave 67%. Finally, using only acoustic observations as input gave a result of 68%.

5.2. Discussion

The first thing to note is that adding real articulatory information to the acoustic data gave a 9% (13% 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 [1].

From the scores of the systems which use only articulation as input, we see that the real articulatory data leads to better classification performance than the simulated, although the difference is not huge. As we have already noted, augmenting the acoustic feature set to include *real* articulatory information gives a performance improvement. At present, there is no gain when instead we add the *recovered* articulatory traces to the acoustics.

One possible explanation as to why the recovered articulation gives no further discriminatory power to the acoustic data, lies with type of networks used to carry out the inversion mapping. The network provides an *averaged* articulatory configuration for each sequence of acoustic observations. Thus, an articulator which shows many different behaviours for one segment type in the original data, is often confined to follow a more uniform set of trajectories in the network output. In this case, articulators appear to be critical more often than they should, and so the ability to place importance on different streams of the data is lost.

6. 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, very similar to that in [6].

6.1. 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 hypothesis, 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 ensures minimum time is used exploring unlikely paths.

The trajectory nature of the LDM means that the state of the system at time t is dependent on its value at time $t = 0$. The practical outcome of this for decoding purposes is that a separate Kalman smoother has to be run to infer a state sequence 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.

Duration modelling is implicit for an LDM, as likelihoods peak at the end of regions which have been explained well by the model. We overlay an explicit duration model in the form of phone-dependent durational probabilities to ensure that suitable phone end-times are assigned higher likelihoods.

6.2. Results and discussion

A preliminary result for the implementation of the decoder for the same task as before, using a feature set comprising acoustics and real articulatory trajectories is a phone accuracy of 56% (Table 2)

feature set	correct	accuracy
real EMA + cepstra + energy	62%	56%

Table 2: Recognition scores from a system built on acoustic and real articulatory data.

The classification score for the same models and feature set is considerably better (76%) than that for recognition (56%) which suggests that the segmentation performance needs to be improved.

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, and 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 [5]. It is expected that Viterbi training will improve performance, as to date the models have been trained using alignments from an acoustic HMM system. These phone boundaries are likely to be different from segmentations produced by LDMs and a combination of articulatory and acoustic gestures.

7. 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 poses 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 better reflect the nature of the data. Co-articulation confounds phone-based systems; however a longer unit based on mixed acoustic and articulatory features would include a certain amount of co-articulation information.

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

Practical speech recognition 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, i.e. the inversion mapping and the set of LDMs, will be trained together.

8. References

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

[2] 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.

[3] 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.

[4] 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.*, vol. 92, no. 2, pp. 688–700, August 1992.

[5] 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*, vol. 1, no. 4, pp. 431–442, October 1993.

[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.