Recognition of voiced sounds with a Continuous State HMM

S. M. Houghton, C. J. Champion and P. Weber

School of Electronic, Electrical and Systems Engineering,
Gisbert Kapp Building, University of Birmingham B15 2TT, UK.
{s.houghton,c.champion}@bham.ac.uk, phil.weber@bcs.org.uk

Abstract

Many current speech recognition systems use very large statistical models using many thousands, perhaps millions, of parameters to account for variability in speech signals observed in large training corpora, and represent speech as sequences of discrete, independent events. The mechanisms of speech production are, however, conceptually very simple and involve continuous smooth movement of a small number of speech articulators. We report progress towards a practical implementation of a parsimonious continuous state hidden Markov model for recovery of voiced phoneme sequences from trajectories of such continuous, dynamic speech production features, using of the order of several hundred parameters. We describe automated training of the parameters using a forced alignment procedure, and results for training and testing on an individual speaker.

Index Terms: continuous state HMM, speech recognition, voiced sounds

1. Introduction

We report progress towards a practical continuous-state hidden Markov model (CS-HMM) for phoneme recognition. Our system requires a very small number of parameters, a few hundred, to model and recover a speech signal encoded as smoothly-varying parameters of speech dynamics. The main goal of our earlier theoretical paper was to introduce the CS-HMM model for speech dynamics [1]. Limited experiments were presented to validate the theory, illustrating the ability of the model to recover sequences of phonemes, using Vocal Tract Resonance data (VTR) [2] for a few TIMIT utterances [3]. Examples were selected with mainly voiced phonemes, and the parameters in the model were manually tuned. In the previous paper we showed that it was critical to have an accurate inventory of phoneme target frequencies for successful recovery.

In the present paper we describe use of the CS-HMM model to automatically train and iteratively refine this phoneme inventory from training data, using a forced-alignment procedure. We also describe extensions to incorporate a language model and formant amplitude information. In the second part of the paper we describe experimentation and phoneme recovery results in which the model parameters are trained and tested over restricted subsets of the TIMIT VTR data. This is the first time that work describing automatic training of this type of system has been reported.

The CS-HMM model is inspired by the Holmes-Mattingly-Shearme (HMS) model of speech [4], and early work in speech recognition. In these models, articulation is modelled as a series of target frequencies (dwell phases) connected by transitions. This is motivated by the conceptual simplicity of the mechanisms of speech production, in which a small number of speech articulators move together to generate all the sounds found in speech. The motion of the articulators is continuous, so a large part of speech consists of smooth motion of the acoustic features from properties defined by one sound, to another. It should therefore be possible to encode this simplicity, using knowledge of speech production mechanisms, in a parsimonious model of speech. Our belief is that such a model is more faithful to the underlying generation of the speech signal and has the potential to be more robust to variation in natural speech. This runs counter to conventional discrete state HMM (DS-HMM) and Deep Neural Network (DNN) speech recognition systems which to an extent ignore the processes of speech generation and use highly complex statistical models to model the variability in speech.

In the next section we provide a short description of the CS-HMM, together with references to further details. The following sections describe the training procedure that we apply and the recognition results that we achieve. The article concludes with some thoughts about how this work might be further extended.

2. CS-HMM theory

The algorithm we use is a sequential branching process to recover a phoneme sequence together with times during which each phoneme is realised. It can be thought of as a variant of the hidden Gaussian Markov model [5, 6]. In this system a state consists of both discrete and continuous components. The discrete components encode the phonetic history to this point (phonemes and timings), the time spent so far in the current phoneme and any other details that may be necessary — the discrete components are represented by $\boldsymbol{d}$. The continuous component is a vector in $\boldsymbol{x} \in \mathbb{R}^m$ of the realised targets in a dwell. In a transition region this is extended with a further m-dimensions to account for the rate of change of formant frequencies. As shown later in this paper, additional continuous components can be included to account for variation in amplitude of the different formant frequencies.

As observations are made, we form a hypothesis of how these observations could be explained. In a hypothesis, information takes the form of a Baum-Welch alpha value, written $\alpha_t(\boldsymbol{x}, \boldsymbol{d})$ where t is time. This value is the sum of path probabilities over all paths arriving in state $\boldsymbol{x}$ at time t, where the paths are limited to those consistent with the discrete state components $\boldsymbol{d}$. Here, a path probability is the product over previous times of the state probability, conditioned on its predecessor, and the observation probability. We are using the term ‘probability’ rather loosely to denote the value of the PDF.

Each hypothesis stores information about an infinite set of continuous components $\boldsymbol{x}$. This requires that the α_t must take a functional form dependent on $\boldsymbol{x}$. We assume that the α_t take the parametric form

$$\alpha_t(\boldsymbol{x}, \boldsymbol{d}) = K_t n\left(\boldsymbol{x} - \boldsymbol{\mu}_t, P_t\right) \tag{1}$$

where for convenience we write

$$n(\boldsymbol{x} - \boldsymbol{\mu}, P) = \sqrt{\frac{|P|}{(2\pi)^m}} \exp -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T P(\boldsymbol{x} - \boldsymbol{\mu}) \tag{2}$$

as the probability density function of a Gaussian distribution with precision matrix P (the inverse of covariance) and expected value $\boldsymbol{\mu}$. The parameters of the Gaussian distribution, that is the mean $\boldsymbol{\mu}_t$, precision P_t and scale factor K_t are stored as part of the hypothesis. The scale factor K_t is the sum of probabilities of all paths consistent with the given hypothesis and is taken as the score of the hypothesis. Whenever hypotheses should be pruned, or thresholded (as described below), K_t is the correct quantity to use for this.

We initialise a hypothesis for each potential initial phoneme, indexing the phonemes by i,

$$\alpha_0^{(i)}(\boldsymbol{x}) = n(\boldsymbol{x} - \boldsymbol{f}_i, A)\mathbb{P}(i) \tag{3}$$

where $\boldsymbol{f}_i$ are the canonical formant frequencies for phoneme i, taken from the phoneme inventory, and these are realised with precision P. The hypotheses are weighted by the likelihood under some language model of the utterance beginning with phoneme i, this is the probability $\mathbb{P}(i)$.

In this work we investigate the case of a flat language model, where all phonemes are equally likely and all transitions between different phonemes are equally likely (repeated phonemes are forbidden). A bigram language model is also considered.

The canonical frequencies and precision A can be learned from data, see section 3. The initial α's are of the form in eq. (1) and the inductive nature of the calculation ensures that this form is maintained.

The modelling assumption is that voiced sounds are represented by a dwell phase during which the acoustic features are constant. A dwell phase is connected to the next by linear transition. The onset of a transition is assumed synchronous across all acoustic features.

2.1. Stepping through a dwell

Suppose the system is in a dwell state at time t–1, that the dwell state has persisted for $h-1$ time-steps and observation $\boldsymbol{y}_t$ is made. Then, assuming Gaussian measurement errors, with precision E, the observation is drawn from the PDF $n(\boldsymbol{y}_t - \boldsymbol{x}, E)$. The hypothesis can be updated to take account of the observation

$$\begin{aligned} \alpha_t(\boldsymbol{x}) &= K_{t-1} n(\boldsymbol{x} - \boldsymbol{\mu}_{t-1}, P_{t-1}) n(\boldsymbol{y}_t - \boldsymbol{x}, E) && (4) \\ &= K_t n(\boldsymbol{x} - \boldsymbol{\mu}_t, P_t) && (5) \end{aligned}$$

where $K_t, \boldsymbol{\mu}_t$ and P_t can be found be expanding and completing the square [1]

$$\begin{aligned} P_t &= P_{t-1} + E, \\ P_t \boldsymbol{\mu}_t &= P_{t-1}\boldsymbol{\mu}_{t-1} + E\boldsymbol{y}_t, \\ K_t &= K_{t-1}\sqrt{\frac{|P_{t-1}||E|}{|P_t|(2\pi)^m}} \\ &\quad \times \exp -\frac{1}{2}(\boldsymbol{\mu}_{t-1}^T P_{t-1}\boldsymbol{\mu}_{t-1} + \boldsymbol{y}_t^T E \boldsymbol{y}_t - \boldsymbol{\mu}_t^T P_t \boldsymbol{\mu}_t). \end{aligned}$$

After accounting for the observation, we consider two possibilities. Either the system continues in a dwell state, or it has reached the end of the dwell and a transition should begin. It is straightforward to see that

$$\begin{aligned} \mathbb{P}(\text{stay in dwell}) &= \mathbb{P}(\text{dwell time} \geq h+1 | \text{dwell time} \geq h) \\ &= \frac{\mathbb{P}(\text{dwell time} \geq h+1)}{\mathbb{P}(\text{dwell time} \geq h)}. \end{aligned} \tag{6}$$

Thus, provided $\mathbb{P}(\text{dwell time} \geq h)$ can be found in some way, an arbitrary timing model can be incorporated. The hypothesis is branched with K_t weighted by the probability found in (6) of staying in a dwell, and its complement of beginning a transition. Despite the convenience of applying a general timing model, in this work we restrict attention to a uniform timing model with dwell times drawn from $\mathrm{Unif}[0, d_{\max}]$ and transition times drawn from $\mathrm{Unif}[2, t_{\max}]$. During the training procedure, $d_{\max}$ and $t_{\max}$ will be found from data.

2.2. Transition regions

Stepping through a transition region is performed in an analogous way to stepping through a dwell region, although the update formulae are a little more complicated. For full details of the calculation we refer the reader to [7].

In a transition region the continuous state component is extended with a m-long vector $\boldsymbol{s}$ of the rates of change. When the transition comes to an end, the hypothesis must be reparametrised in terms of the target frequency of the next dwell $\boldsymbol{x}' = \boldsymbol{x} + h\boldsymbol{s}$. Thus, we must marginalise against the slope variables $\boldsymbol{s}$

$$\begin{aligned} \alpha_t(\boldsymbol{x}') &= \int \alpha_t(\boldsymbol{x}, \boldsymbol{s}) d\boldsymbol{s} = \int \alpha_t(\boldsymbol{x}' - h\boldsymbol{s}, \boldsymbol{s}) d\boldsymbol{s} && (7) \\ &= K_t n(\boldsymbol{x} - \boldsymbol{\mu}', P') && (8) \end{aligned}$$

where $\boldsymbol{\mu}'$ and P' must be computed. For full details of the calculation, refer to [7].

2.3. Incorporating a language model

A language model can be easily incorporated at the beginning of each dwell phase. At this point of the recognition process, a hypothesis is parametrised by $\alpha_t(\boldsymbol{x}, \boldsymbol{d})$ and must branch into N hypotheses where N is the total number of possible continuations, this will usually be the number of phonemes being considered. To achieve this,

$$\alpha_t^{(i)}(\boldsymbol{x}, \boldsymbol{d}_i) = \alpha_t(\boldsymbol{x}, \boldsymbol{d}) n(\boldsymbol{x} - \boldsymbol{f}_i, A)\mathbb{P}(i|\boldsymbol{d}) \tag{9}$$

for $i = 1, 2, \ldots N$, where $\boldsymbol{d}_i$ represents the discrete state components extended to include beginning phoneme i, $\boldsymbol{f}_i$ are the canonical target frequencies for phoneme i and these are realised with precision A. Finally, $\mathbb{P}(i|\boldsymbol{d})$ is the *language model* — it is the probability of beginning phoneme i given the phonetic history (and any other information) stored in the discrete state components $\boldsymbol{d}$.

As will be described in section 3, by using a highly constrained language model the system can be forced to perform Viterbi alignment against a known transcript as part of a training procedure. We present results to show the impact of a simple bigram language model on recognition performance.

2.4. Thresholding

Through the branching process, the total number of hypotheses grows exponentially as observations are made. For an efficient

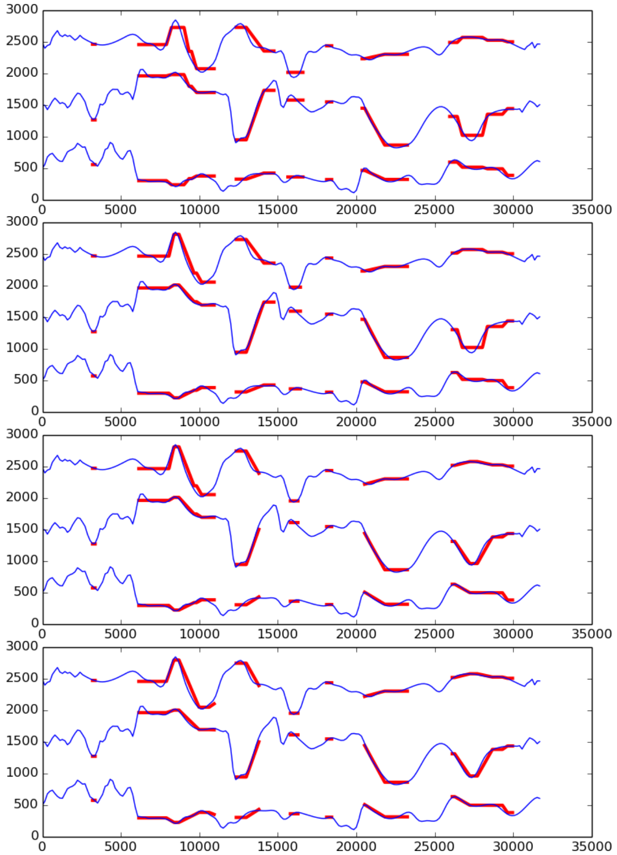

Figure 1: *ah — uw y ih er — l ey — er — ih — ow m — eh l ih n* where the long dash is used to show regions of speech that are not being considered, i.e. unvoiced sounds. VTR tracks (blue) are shown together with Viterbi alignments (thick red) after 0, 2, 5 and 8 iterations of the training process.

implementation we apply two thresholding techniques. First we maintain only a list of the top-N most likely hypotheses and secondly, any hypotheses with $\log K_t < \log \kappa - 100$, where κ is the largest likelihood, is discarded. Typically, $N = 250$ is used. These are approximations and may mean that the overall most likely path is discarded. However, the results presented here are robust to a range of a different thresholding values.

3. Training procedure

In this paper, we take the vocal tract resonance data (VTR) [2], together with TIMIT phonetic transcriptions [3] as our start point. The VTR data was automatically generated and then (partially) hand corrected. This may have a negative influence on our work as some of the tracks are heavily smoothed while others are not. This could lead to some inconsistencies, more discussion of this later. All methods described here apply equally well to any tracks that are believed to follow the dwell transition pattern, for example articulator positions from MOCHA TIMIT [8], and perhaps even the output from a bottleneck layer of a neural network [9].

The TIMIT corpus has hand labelled phoneme boundaries. These do not identify where the dwell regions are within each phoneme — indeed it is difficult to identify dwell regions within some of the TIMIT markings. We find the location of dwell regions through an iterative Viterbi alignment process.

First, we apply a standard mapping to reduce the number of phonemes to 49 [10], and then extract the voiced regions by restricting to the set of phonemes */aa ae ah ao aw ay eh er ey hh ih iy l m n ng ow oy r uh uw w y/* of size $N_\phi = 23$. It is assumed that the TIMIT phoneme boundary at the beginning and end of each voiced region is marked correctly. Upon checking some of these, this appears a good assumption — the boundary between voiced and unvoiced is usually well defined, but not always synchronous across the frequency range.

An initial phoneme inventory is built by assuming that the TIMIT boundaries mark the beginning and end of dwell phases. We know this to be false, as by our modelling assumptions there must be transitions, but it gives a start point. A forced alignment is then performed on each voiced region in turn. This is essentially recognition but with a language model that only permits phonemes to occur in the correct order. The most likely hypothesis to explain the full phoneme sequence is reported, along with the timing of each dwell. This allows a refined phoneme inventory to be produced based only on the sections reported as dwell regions, and the process iterated.

As mentioned above, to prevent the number of hypotheses growing too large some type of thresholding must be applied (see section 2.4). We find that during the Viterbi alignment this must be handled very carefully to ensure that the *correct* explanation of the data remains in the list of hypotheses being considered. This is resolved by retaining a hypothesis for every (phoneme, time since beginning of previous dwell) pair. Simple calculation then shows that the maximum number of hypotheses that might be retained is $N_\phi(d_{\max} + t_{\max})$.

Transition regions are ignored for the purposes of estimating the measurement precision E. This combined with re-estimating the start and end of each dwell region on each iteration means that the actual data being used for parameter estimation changes from iteration to iteration. As will be seen, this can lead to a degradation in performance on some iterations. A single measurement precision matrix E is computed across all the data. It seems reasonable to neglect the transition regions when estimating E as human listeners are much less sensitive to formant frequencies during transitions than during dwells [11]. It is reasonable to assume that the range of realisations for different phonemes will be different, however, given the limited data the realisation precision A has been pooled across phonemes.

An example of the alignment process is shown in figure 1. This is sentence SI731, “A few years later the dome fell in”, spoken by TIMIT speaker MWEW0 from dialect region 2. This sentence has 7 voiced regions, as given in the caption to figure 1. Looking at F_2 in the final voiced section (around sample 27000) it is clear to see how the alignment process has resulted in a model that much better fits the data, identifying a dwell region with transitions to and from it. Another clear example is F_3 at around 8000 samples into the utterance.

4. Initial results

In this work, we restrict attention to a single TIMIT speaker, MWEW0 from dialect region 2, and exclude the SA sentences. Given the small amount of data, we train our system on the eight utterances and then test on each of the utterances. After each iteration of the training procedure, we perform a recognition experiment. The results are shown in table 1. The impact of the alignment process can be seen. We have a large reduction in deletions and increase in the number of phonemes being recognised correctly. However, rather disappointingly the overall er-

Iteration	Corr	Subs	Deln	Inst	Err
0	27.4	31.7	40.9	0.0	72.6
1	29.3	33.5	37.2	0.0	70.7
2	32.9	32.9	34.1	0.0	67.1
3	34.1	32.9	32.9	0.0	65.9
4	32.9	35.4	31.7	0.0	67.1
5	33.5	36.0	30.5	1.8	68.3
6	32.9	35.4	31.7	2.4	69.5
7	34.1	34.8	31.1	2.4	68.3
8	36.0	34.1	29.9	2.4	66.5
9	34.8	36.0	29.3	2.4	67.7
10	34.1	36.6	29.3	3.0	68.9

Table 1: Recognition performance after iterations of the alignment procedure. Here, alignment and recognition are based purely on the VTR data for voiced regions.

ror rate does not show a dramatic reduction. After 10 iterations of the alignment process the system has converged. There are 85 trainable parameters in this system.

In these initial results we have used a flat language model, and made no use of amplitude information. Making use of an improved language model or amplitude information would be expected to lead to an improvement in performance: we consider them in the following sections.

5. Language model

In section 2.3 we discussed how to incorporate a language model into the recognition system. We learn the language model from the full TIMIT training set (excluding SA sentences). We use a simple unigram model for the initial phoneme of each voiced segment and a bigram model for the sequence of phonemes.

6. Amplitudes

The VTR data does not include amplitude data, however, as the data is all derived from TIMIT and we have the audio files then amplitudes can be computed. Spectral amplitudes are computed on 10ms windows of audio and then a quadratic interpolation applied to find amplitude at the required frequency. The feature used for the system is log-amplitude as this better fits the Gaussian assumption.

All theory presented in section 2 considered feature vectors of formant frequencies. Log-amplitude features are appended to these vectors, and all the theory follows through. We do not permit any correlation between formant frequencies and their log-amplitudes. A final set of experiments are performed with log-energy rather than log-amplitude. Energy is taken to be the sum of the squared sample values in a 10ms frame. We do not use any tapered windowing functions, nor do the frames overlap.

7. Results with extended acoustic features

Recognition performance with the inclusion of log-formant-amplitudes, log-energy and a bigram language model is shown in table 2. In all cases, we see overall error rates decline as the alignment process is carried out. We also see improvements when additional features are used and substantial improvements in overall performance through the addition of a language model.

Features	Iteration	Corr	Subs	Deln	Inst	Err
F	0	27.4	31.7	40.9	0.0	72.6
F	10	34.1	36.6	29.3	3.0	68.9
F+A	0	36.0	30.5	33.5	3.7	67.7
F+A	10	45.1	33.5	21.3	9.8	64.6
F+E	0	33.5	27.4	39.0	0.0	66.5
F+E	10	40.9	30.5	28.7	4.3	63.4
F+LM	0	34.1	24.4	41.5	0.6	66.5
F+LM	10	39.6	31.1	29.3	2.4	62.8
F+A+LM	0	43.9	25.0	31.1	1.8	58.0
F+A+LM	10	51.2	30.5	18.3	8.5	57.3
F+E+LM	0	38.4	22.6	39.0	0.0	61.6
F+E+LM	10	50.6	23.2	26.2	2.4	51.8

Table 2: Recognition performance for different parametrisations, before and after the alignment process. Here, F are formant frequency features, A are log-formant-amplitudes, E is log-energy and LM shows when a bigram language model has been used.

It is perhaps surprising to see that an overall lower error rate is achieved when adding the single extra feature of log-energy compared to the 3 extra features of log-formant-amplitudes. This difference is marginal when no language model is applied, but much clearer to see when a language model is applied. The cause of this is not known for certain, but our feeling is that the occasional departures of the formant labels from the true formant trajectories have inconsistent effects on the amplitude estimates derived from them. While the additional features should be useful, they are in fact too noisy and so lead to a degradation of the system. On the other hand, the energy of the signal is computed from the squares of sample values and so is robust against any errors in the VTR data.

The best performance achieved with this system sees a correct rate of 50.6% and a phoneme error rate of 51.8%. The parameter count for this particular system is 110 to model the acoustics and 529 for the language model. This system has been training with an automatic procedure on less than 20 seconds of data. We have not applied any language model scale factors, or word insertion penalties, as are often found in conventional HMM-based recognisers.

8. Summary

In this work we have presented a system to recognise voiced sounds. The system is automatically trained on a very small amount of data through a Viterbi alignment process. This greatly improves on previous work where the system was hand-tuned on just a few sentences.

There are a number of improvements to make before we have a complete speech recognition system based on a CS-HMM. One area for investigation is how to link this system with one designed for the unvoiced sounds (see [12]). Further work is then to expand this system so that it can be applied across multiple speakers. This will likely require the inclusion of a context-dependent phoneme inventory. We believe that these challenges can be met without vastly increasing the number of parameters in the system.

9. References

[1] P. Weber, S. M. Houghton, C. J. Champion, M. J. Russell, and P. Jančovič, "Trajectory analysis of speech using continuous state hidden Markov models," in *ICASSP*, 2014.

[2] L. Deng, X. Cui, R. Pruvenok, J. Huang, S. Momen, Y. Chen, and A. Alwan, "A database of vocal tract resonance trajectories for research in speech processing," in *ICASSP*, 2006.

[3] J. S. Garofolo, "TIMIT: Acoustic-phonetic continuous speech corpus," Linguistic Data Consortium, Tech. Rep., 1993.

[4] J. Holmes, I. Mattingly, and J. Shearme, "Speech synthesis by rule," *Language and Speech*, vol. 7, pp. 127–143, 1964.

[5] P. L. Ainsleigh, "Theory of continuous-state hidden Markov models and hidden Gauss-Markov models," Naval Undersea Warfare Center Division (Newport, Whode Island), Tech. Rep. 11274, 2001.

[6] P. L. Ainsleigh, N. Kehtarnavaz, and R. L. Streit, "Hidden Gauss-Markov models for signal classification," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1355–1367, 2002.

[7] C. Champion and S. M. Houghton, "Application of continuous state hidden Markov models to a classical problem in speech recognition," *Computer Speech and Language*, submitted, 2015.

[8] A. Wrench, "MOCHA-TIMIT," http://www.cstr.ed.ac.uk/research/projects/artic/mocha.html, 1999.

[9] L. Bai, P. Jančovič, M. Russell, and P. Weber, "Analysis of a low-dimensional bottleneck neural network representation of speech for modelling speech dynamics," submitted to *Interspeech*, Dresden, Germany, 2015.

[10] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[11] W. A. Ainsworth, "Perceptual tolerance of the shape of formant transitions," *Proceedings of the Institute of Acoustics*, vol. 18, no. 9, pp. 67–74, 1996.

[12] P. Weber, C. J. Champion, S. M. Houghton, P. Jančovič, and M. J. Russell, "Consonant recognition with CSHMMs and perceptually motivated features," submitted to *Interspeech*, Dresden, Germany, 2015.