Application of Continuous State Hidden Markov Models to a Classical Problem in Speech Recognition

Colin Champion*, S. M. Houghton*

School of Electronic, Electrical and Systems Engineering, University of Birmingham, Gisbert Kapp Building, University of Birmingham B15 2TT.

Abstract

This paper describes an optimal algorithm using continuous state Hidden Markov Models for solving the *HMS decoding problem*, which is the problem of recovering an underlying sequence of phonetic units from measurements of smoothly varying acoustic features, thus inverting the speech generation process described by Holmes, Mattingly and Shearme in a well known paper (*Speech synthesis by rule*, Language and Speech **7** (1964)).

Keywords: Speech Recognition, Hidden Markov Model, Recognition by Synthesis.

1. Introduction

1.1. Overview

This paper addresses the problem of correctly incorporating dynamic information into the acoustic models used for speech recognition.

For several decades the dominant algorithms in the field have had recognised weaknesses in handling dynamics. The algorithms are based on Hidden Markov Models (*HMM*s) in which the state space is discrete – for this reason we will refer to them as Discrete State HMMs (or *DS-HMM*s). The feature vectors have been made up of spectral band energies or their transformation into cepstra. An overview of the use of HMMs in speech recognition is given by Gales and Young (2007).

The physical properties underlying speech consist of the smooth motion of articulators between positions defined by the various sounds. The same smoothness can be seen in acoustic features – at least for sonorant sounds – if they are chosen appropriately whereas features chosen for their ease of extraction may exhibit intractable dynamics.

*Corresponding author

Email addresses: c.champion@bham.ac.uk (Colin Champion), s.houghton@bham.ac.uk (S. M. Houghton)

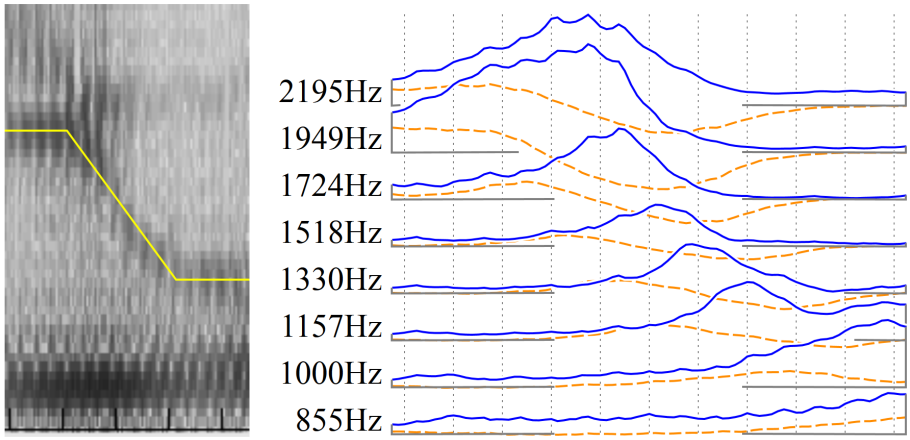

Figure 1: Spectrogram showing the first two voiced phonemes of 'He will...' together with band energies and their derivatives for the spectral region containing F2.

Models of speech which preserve the salient features of the production process are attractive for use in recognition because they inherit the smoothness of the underlying mechanisms. Conventional recognisers fail in this respect for two reasons. Firstly a continuous transition cannot be properly modelled as a sequence of discrete states, and secondly the approximately linear properties of the underlying features lose their structure when expressed in terms of spectral band energies.

The designers of speech recognition systems have sought to remedy these weaknesses by a number of strategies. One is to incorporate time derivatives (deltas) of cepstral features (Furui, 1981). This is easy to accomplish but questionable in terms of model coherence (Tokuda et al., 2003).

Figure 1 shows the spectrogram for part of the TIMIT utterance 'He will allow a rare lie'. It contains most of the steady state (for which we will later define the term *dwell*) of the 'e' of 'he', the transition to the following 'w', and most of the corresponding steady state. The F2 path (idealised as a piecewise linear yellow line) is fully intelligible in terms of the phonetic sequence. The solid blue lines on the right of the figure show the energies of triangular mel-spaced bands in the F2 region. Each of these makes sense in terms of the formant motion but not phonetically; it seems odd, for instance, to say that the transition from 'e' to 'w' is characterised by a brief activation of the band at 1724Hz about an eighth of the way through. The linearity of the formant feature has been replaced a complicated interdependence between band energies.

The orange dashed lines are the derivatives of the band energies (computed in the normal way).

The second strategy for capturing dynamic properties in a speech recogniser is to add parameters whose function is to describe transitions explicitly. We

contend that it is more natural and more economical to infer the properties of transitions through the structure of the model. This is the subject of the present paper, in which we seek to show how models of individual phonetic units can be constructed in such a way that the properties of transitions can be inferred by interpolation. A model with these properties is likely to be closely related to the speech production process (as ours is), since acoustic features gain their interpolability by reflecting the smooth motion of articulators.

Standard DS-HMMs try to model transitions by splitting each sound into substates some of which fall in transition regions, and then estimating the parameters of each phonetic unit according to its context. If the transition from φ_0 to φ_1 is split into sufficiently many substates, and if the properties of each substate are estimated for the pair (φ_0, φ_1) rather than for an individual phonetic unit, then a good enough approximation will be obtained.

The main penalty to capturing properties through parameters rather than through model structure is that the amount of training data needed by an algorithm increases with the size of its parameter space. But a more insidious penalty lies in the fact that an algorithm whose strength lies in the number of its parameters sheds no light on the problem it addresses.

1.2. Views of speech dynamics

A number of attempts have been made to incorporate faithful models of speech dynamics into recognition algorithms. Many of these have been based on segment models whose theory was developed by Gales and Young (1993), by Russell (1993), by Holmes and Russell (1999), and by others.

If the decision is taken to view speech as comprising trajectories in a suitable space, then a question arises over the position in the model of the variables subject to trajectory dynamics. Two options suggest themselves. The first is to construct models with hidden components behaving in a smooth way and determining the observation probabilities (which may be based on power spectral estimates). A mapping needs to be constructed from the states to the observations, and this is where the difficulty lies: if the state space is based on anything like formants and the observations are anything like spectral bin energies, then the true relationship between them is radically non-linear, and any approximation which lends itself to analytical treatment (such as assumed linearity) will be highly counterfactual.

Examples of this approach are the methods developed by Deng and Ma (Deng, 1998; Deng and Ma, 2000), and the Multiple-level Segmental HMMs of Russell and others (Russell and Jackson, 2005; Russell et al., 2007). Russell and his co-workers assume a linear mapping as an approximation to the relationship between spectral measurements and the underlying features; the remaining papers invoke multilayer perceptrons to overcome the nonlinearity of a relationship which is not further specified.

The second avenue is to directly estimate the underlying smoothly varying features from the audio signal and to use them as input to a recognition algorithm. The most significant difficulty here lies in feature estimation: if the

features are interpreted as formant frequencies then we immediately encounter the problem of formant tracking, which can be performed fairly reliably by eye but for which no satisfactory computer algorithms exist (Deng et al., 2006). Direct measurement of articulatory features, eg. by EMA (Richmond et al., 2003; King et al., 2007), would be an alternative approach, but here too there are difficulties in estimating the features directly from audio.

The second difficulty lies in making use of the measured features. This is the problem addressed below: models with hidden components need to address the same problem, so in one form or another it has received considerable attention in the literature.

Most earlier work has been based on segmental HMMs, whose relationship to the algorithm of this paper will be made clear later (see §2.4): we claim that we are providing an optimal solution to a problem which is only approximately solved by segmental HMMs.

Another related approach is the Hidden Dynamical Model of Richards and Bridle (1999). This uses a characterisation of formant tracks which is more flexible than piecewise linearity (and may therefore be said to model the smooth acceleration as well as the smooth motion of articulators), but is able to do so only at the cost of not being expressible in a computationally feasible algorithm for recovering the optimal phonetic or lexical sequence. The Hidden Dynamical Model can be used to rescore a recovery made by other means but not as the basis for a decoder in its own right. We believe that the method of the present paper does the best job possible of capturing the properties of speech which arise from its production method subject to the constraint that its optimum can be found by an efficient search algorithm (in our case the dynamic program).

Less directly comparable are the methods adopted by recognisers based on neural networks. These contain nothing specific to dynamics at all: they see nothing but a sequence of observations. If an exploitable structure is apparent from the training data then they may take advantage of it. Having no concept of state, a feed-forward neural network is limited to what it can see within a given time window, but recurrent neural networks can make use of cyclic connections to capture longer term effects (see Sak et al. (2014)).

1.3. A model of speech

The starting point of the present paper is the synthesis model put forward by Holmes et al. (1964, hereafter *HMS*), which portrays speech as an alternation between dwells at phonetic targets and transitions between them, and in which the motion is roughly linear in a suitable space, understood by the authors as the space of formant frequencies. Holmes, Mattingly and Shearme did not propose a recognition algorithm based on their model, though John Holmes repeatedly advocated use of parametrically modelled dynamics for speech recognition (Holmes and Holmes, 2001), an approach which has come to be known as *Recognition by Synthesis* (Paliwal and Rao, 1982).

This paper assumes a simplified version of the HMS model. Each phonetic unit is characterised by a *canonical target* for the acoustic feature. We think of

formant frequencies as the prototypical features but any measurable properties (such as loudnesses or bandwidths) can be included. An articulation of a sound includes a period (the *dwell*) during which the *realised target* is constant: the realised target comes from a Gaussian distribution whose mean is the canonical target. Observations will be distributed (and assumed Gaussian) about the realised target. Dwells will normally be assumed to have positive length, but dwells of length 0 are covered by the same formulae (see §2.9) whereas dwells of negative length (explained in §2.10) can be handled with a little extra work.

The articulators move smoothly between one phonetic unit and another: the intervening period is known as a *transition*, in which the acoustic feature starts from the realised target of one unit and moves linearly to the realised target of the next. Transitions of length 0 (which would equate to discontinuities in the trajectories) are prohibited *a priori* although they would not be difficult to allow for (and might be appropriate for modelling unvoiced sounds).

The smooth motion assumed by our model is in sympathy with the segmental properties of the models cited in §1.2. We simplify the HMS model by concentrating our attention on sonorant sounds, and by treating a transition as a single linear trajectory rather than as two.

1.4. Realised and canonical targets

The distinction between realised and canonical targets is absent from HMS, which seeks to specify an exemplar of a sound rather than the space of legal instances.

If we consider a single occurrence of a phonetic unit, there will be systematic departures of the observations from the canonical mean: these are reflected in the displacement of the realised from the canonical target. The causes will include factors due to the dimensions of the speaker's vocal tract and to other properties of his physiology and behaviour; due to phonetic phenomena such as coarticulation; and due to random imprecisions in motor control. A more complete model of speech would handle some of these factors explicitly. We discuss later (§4.1) how the model of the present paper extends to vocal tract length; coarticulation also fits naturally into the same framework.

All non-systematic departures of observations from the canonical target are treated as Gaussian noise about the realised target. The two forms of departure from the canonical norm are illustrated in Figure 2.

Although the departure of observations from the realised target is treated as a Gaussian error term, it should not be seen as a satisfactory representation of measurement error. A main component will be the departure of formant tracks from piecewise linearity.

On the other hand the most important source of measurement error, so far as a formant tracker is concerned, is likely to lie in labelling. We do not know the limit of what can be achieved by a formant tracker, but we can be fairly certain that there will be cases in which the acoustics do not determine whether a spectral peak corresponds to F2 or to F3. If errors of this sort are present they will be severely non-Gaussian. However the natural way to guard against them

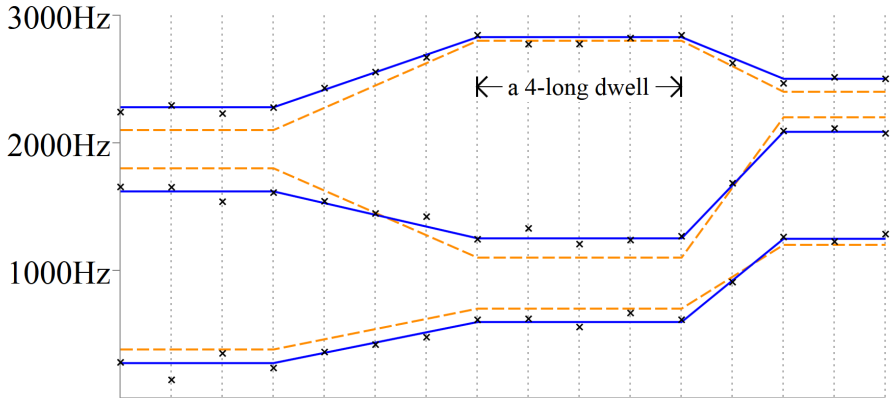

Figure 2: An idealised model of formant tracks. The canonical pronunciations are shown as dashed orange lines; the realised targets (solid blue) differ slightly from them; and the observations are crosses distributed about the realised targets.

is by requiring the formant tracker to output alternative labellings whenever there is ambiguity (as does the formant tracker described by Holmes (2001)). The decoding algorithm can then consider alternative labellings at the same time as it considers alternative phonetic hypotheses.

1.5. The continuous state HMM and other formalisms

The method of this paper is based on the *CS-HMM* algorithm (standing for 'continuous state HMM'). CS-HMMs have been known about for some years (Ainsleigh, 2001). The operations available for DS-HMMs – the dynamic program (Bellman, 1954), the alpha and beta passes, the gamma calculation, and the re-estimation formulae (Baum et al., 1970) – all carry over to CS-HMMs given suitable assumptions of Gaussianness and linearity.

The proofs are fairly direct. At each step a term is derived which is the product of scaled Gaussians; the formulae are expanded (perhaps at considerable length) and then collapse down (by the process of completing the square) to give a new formula which is often very simple. Indeed, it may turn out to be deceptively simple. Sometimes it is the formula that would be guessed by someone who was asked to provide an estimate of a particular unknown from the data seen so far, and this may create the illusion that the unknown is indeed being estimated when in truth it is being marginalised – that is, averaged over.

In our development all distributions except timing are pure Gaussians. The CS-HMM framework allows GMMs to be used in their place (as for DS-HMMs), but we consider them to be inconsistent with our aim of limiting parameters to those which are phonetically meaningful.

CS-HMMs are an alternative formulation of the ideas which are also familiar in the guise of the Kalman filter (Kalman, 1960). The relationship is discussed in detail by Ainsleigh (2001). It is most important, in the current context, to

distinguish the Baum-Welch alpha pass (which computes a sum) from a branching algorithm along the lines of the dynamic program (which finds a maximum). The algorithm of this paper makes sequential assumptions about the discrete state components (seeking the sequence which maximises likelihood), and does so using an alpha pass (and therefore summing paths over continuous components) to compute the likelihoods of the sequences.

1.6. Outline of the present paper

The problem we address is that of recovering an underlying sequence of phonetic units given acoustic measurements: we refer to this as the *HMS decoding problem*. The method we propose is a thresholded dynamic program based on continuous state HMMs, and we believe that it provides a near-optimal recognition procedure. The algorithm is specified in detail by separate equations for the different cases which arise from dwells, transitions and the alternation between them.

A more detailed and precise statement of the algorithm can be obtained from the source code of our implementation (Houghton, 2014).

The claims we make for our algorithm, to be precise, are that it assigns the true probability under the model to each hypothesis it considers, where this probability takes account of all legal piecewise linear continuous trajectories; that hypotheses are ranked according to their likelihoods at each point in a way which sacrifices little information; and that the likeliest path will be output at the end so long as the thresholding has not been too aggressive. We also believe that the algorithm is as computationally efficient as the nature of the optimisation problem allows.

Our discussion does not cover parameter estimation. We have already mentioned the fact that the Baum-Welch algorithm can be applied to CS-HMMs as well as to DS-HMMs, but we believe that this is more powerful machinery than is needed. The decoding algorithm presented below can be extended to provide a Viterbi training procedure with very little effort, and given our intention to limit the parameters of the system to those which are phonetically meaningful we believe that no more is needed. (In fact Viterbi training is found sufficient for Kaldi by Povey et al. (2011) for parameters other than those relating to GMMs.)

It may be helpful to indicate where the model of the present paper lies in the continuum between 'knowledge-based' and 'data-driven' approaches. The model is automatically trainable but leaves space for the incorporation of linguistic properties felt to be important. It needs sonorant sounds to be modelled differently than unvoiced consonants (although we believe that the right way of analysing unvoiced consonants would have mathematical similarities to the model proposed below).

After describing our algorithm we present an experiment on toy data made up of sinusoidal speech. This experiment shows the effectiveness of the algorithm even when its features are formant estimates from a simplistic tracker. We compare the CS-HMM with the discrete alternative and we compare formant features with cepstra.

We then show how the CS-HMM can be applied to global properties of the speech signal, with vocal tract length being used in illustration. We believe that this shows our model to have a power which others lack. Finally we present a few conclusions.

Time in this paper is discrete (written t – the units of time are *ticks*), and acoustic features are m-dimensional. The phonetic units will be chosen as meaningful categories corresponding to sounds whose distribution can reasonably be modelled as Gaussian. Thus the English schwa would probably belong to the inventory as a single unit even if its instances differ too much to be labelled as a single phone; the consonant 'l' on the other hand has two distinct realisations (clear and dark) which would probably be represented as separate phonetic units.

The phonetic inventory is assumed known and fixed, with the canonical pronunciation of each unit specified by an m-long vector of target values. Durations can be modelled in whatever way is most suitable, and a language model can be used to assist recovery (otherwise we assume that all units are equally likely in all contexts).

2. Application of CS-HMMs to speech recognition

2.1. Structure of the algorithm

The algorithm we describe is a sequential branching process for recovering an unknown phonetic sequence. At each time step we have a list of hypotheses containing information about the state. A state contains both continuous and discrete components. The discrete components are as follows:

- The identity of the current phonetic unit (or of the unit we are leaving in the case of a transition);
- The number of ticks h we have spent in the current dwell or transition;
- The identity of as many previous phonetic units as are needed by the language model; and
- A flag to indicate whether we are in a dwell or a transition.

During a dwell the state has a single continuous component, namely

- an m-long vector comprising the realised targets of the current phonetic unit (where m will be 3 if our features are three formant frequencies).

During a transition the state contains two continuous components, which are

- an m-long vector comprising the realised targets of the phonetic unit just left; and
- an m-long vector of slopes of the acoustic features.

Each hypothesis stores probability information about an infinite set of continuous states in the form of a Baum-Welch alpha value: that is, the alpha value α_t associated with a given state is a sum of probabilities over all paths leading to the continuous state component and consistent with the discrete state history. This is a sum over paths of the product of path transition probabilities and observation probabilities given the path. (We will qualify this slightly when we come to the section on entering a dwell state.) α_t combines information from observations $y_0 ... y_{t-1}$.

In order to be able to store an infinite number of values we need them to take a functional form, and we find that we may express the alpha values of all states associated with a hypothesis in the form of a scaled Gaussian distribution which we shall write as $\alpha_t(\boldsymbol{x})$ during a dwell and as $\alpha_t(\boldsymbol{x}, \boldsymbol{s})$ during a transition, where $\boldsymbol{x}$ is the realised target and $\boldsymbol{s}$ is the slope.

The Gaussianness of the alphas is established inductively from the Gaussianness postulated for the realised targets given phonetic identity and for the observations about the realised targets.

When we express alpha values as a scaled Gaussian, its parameters – the mean, variance and scale factor – are stored as part of each hypothesis. The scale factor is the sum over all paths up to the current time, and ending in the current discrete state, of the joint probabilities along the path: this is exactly the quantity we will want to threshold on in order to limit the number of hypotheses we retain (where 'joint probability' is understood to mean 'product of pdf values').

Hence we may enumerate the components of a hypothesis as follows:

- The discrete state components as listed above; and
- The mean $\boldsymbol{\mu}_t$, precision p_t and scale factor k_t of the scaled Gaussian distribution governing the alphas of all states whose discrete components are those of the current hypothesis.

At each step we will have a number of hypotheses under consideration, each of which will be capable of being extended in more than one way, giving a longer list of candidate hypotheses at the next time step. But as we extend the hypotheses we gain further information from a new observation, allowing us to threshold the list.

If a hypothesis is in a dwell state, we have the choice of continuing the dwell for one more unit of time, or of setting off on a transition, padding out the continuous state with a slope vector.

If a hypothesis is in a transition, we have the choice of continuing the transition for one more tick or of entering a dwell. A third option arises owing to the possibility of zero-length dwells: we may come to the end of one transition and begin another. This option does not need special treatment so long as we are careful in handling both the beginnings and the ends of transitions. The formulae needed for evolving the state in the various cases will be presented below.

As a notational convenience, we will write

$$n(\boldsymbol{x}, p) = (2\pi)^{-m/2}|p|^{1/2}\exp\{-\tfrac{1}{2}\boldsymbol{x}p\boldsymbol{x}^{\mathsf{T}}\} \tag{1}$$

for the Gaussian probability density function in which p is the precision (inverse variance). At no point do we assume precisions to be diagonal, although they will often be so and calculations can be simplified accordingly. Vectors are rows.

With this notation we can represent an alpha value as

$$\alpha_t = k_t n(\boldsymbol{x} - \boldsymbol{\mu}_t, p_t). \tag{2}$$

2.2. Timing model

We define the length (or duration) of a dwell or transition to be the time difference between its end and its start. The number of observations corresponding to it will be 1 greater; hence the dwell near the middle of Figure 2 is 4 ticks long although 5 observations are centred on its target.

We write $P(h)$ for the probability that the next time step stays in a dwell whose length so far is h ticks (with $P(-1)$ for now defined to be 1). The probability of leaving a dwell having spent h ticks in it is $(1 - P(h))$, and we similarly write $P'(h)$ as the probability of staying in a transition given that h ticks have been spent in it. For dwells (though perhaps not for transitions) it is likely to be beneficial to make the timing model depend on the phonetic unit (as $P_\varphi(h)$). The probability of a dwell having total length L is then calculated as

$$(1 - P(L)) \prod_{t=0}^{L-1} P(t). \tag{3}$$

The timing model will be estimated from data. If we make a histogram $D(t)$ of dwell times t, then

$$P(t) = \frac{\sum_{\tau=t+1}^{\infty} D(\tau)}{\sum_{\tau=t}^{\infty} D(\tau)}. \tag{4}$$

The transformation from D to P puts the timing model in the form of a Markov process.

In an exponential timing model the $P(t)$ will be equal, and it is convenient to constrain a model to this form for large durations.

2.3. Stepping through a dwell

Let us begin the induction assuming that the first time step is the start of a dwell period for some unspecified phonetic unit. We will therefore initiate the list of hypotheses with a single entry for each unit in the inventory. For a given φ we shall assume that target realisations $\boldsymbol{x}$ are distributed as

$$n(\boldsymbol{x} - \boldsymbol{\theta}_{\boldsymbol{\varphi}}, c_\varphi) \tag{5}$$

where $\boldsymbol{\theta}_{\boldsymbol{\varphi}}$ is the canonical mean for φ and c_φ is the precision. Meanwhile an observation $\boldsymbol{y}$ is distributed around the unknown realised target $\boldsymbol{x}$ as

$$n(\boldsymbol{y} - \boldsymbol{x}, e) \tag{6}$$

where e is the measurement precision.

Let $\boldsymbol{y}_0$ be the first observed measurement. Then we may write

$$\alpha_1(\boldsymbol{x}) = P(-1)\, n(\boldsymbol{y}_0 - \boldsymbol{x}, e)\, n(\boldsymbol{x} - \boldsymbol{\theta_\varphi}, c_\varphi) = k_1\, n(\boldsymbol{x} - \boldsymbol{\mu_1}, p_1) \tag{7}$$

where

$$\mu_1 = (\boldsymbol{\theta_\varphi} c_\varphi + \boldsymbol{y}_0 e)\,(c_\varphi + e)^{-1}, \tag{8}$$

$$p_1 = c_\varphi + e, \quad \text{and} \tag{9}$$

$$k_1 = P(-1)\, n(\boldsymbol{y}_0 - \boldsymbol{\theta_\varphi}, (c_\varphi^{-1} + e^{-1})^{-1}) \tag{10}$$

and $P(-1) = 1$ by convention. We may set h to 0 for the next step.

The fact that the product of two Gaussians is itself a scaled Gaussian is well known. The process of finding the coefficients of the product is an example of completing the square.

We may proceed in the same way for as many steps as we assume we spend in the dwell phase. The general formula is

$$\alpha_t(\boldsymbol{x}) = k_t\, n(\boldsymbol{x} - \boldsymbol{\mu_t}, p_t) \tag{11}$$

where

$$\mu_t = (\boldsymbol{\mu_{t-1}} p_{t-1} + \boldsymbol{y}_{t-1} e)(p_{t-1} + e)^{-1}, \tag{12}$$

$$p_t = p_{t-1} + e, \quad \text{and} \tag{13}$$

$$k_t = k_{t-1}\, P(h)\, n(\boldsymbol{y}_{t-1} - \boldsymbol{\mu_{t-1}}, (p_{t-1}^{-1} + e^{-1})^{-1}). \tag{14}$$

We notice that the precision p_t increases at each step, implying that the distribution on the realised target becomes tighter as we acquire further observations.

2.4. Relationship to segmental models

It is useful to take stock at this point and see what we have achieved. We have computed the posterior distribution and path likelihood of all possible realised targets $\boldsymbol{x}$ using a hierarchical model in which observations are subject to a Gaussian distribution about the realised target and the realised target is subject to a Gaussian distribution about the canonical target. This is a perfectly routine calculation, but what is interesting is that we have performed it inductively using a Baum-Welch alpha pass iteration.

The main aim of trajectory segmental HMMs is to apply the same hierarchical model to the same sort of observation; however the continuous-valued realised target is excluded from the state. This omission is crucial. In an HMM the state is constrained by a probability distribution as it advances from time to time, and this distribution can be used to enforce continuity. The observations, by contrast, are conditionally independent given state. Therefore continuity can be imposed only on state variables, and when consistency is required of continuous quantities, a CS-HMM needs to be used.

The method by which segmental HMMs seek to impose continuity is to group together the time steps belonging to a dwell period into a single segment. The observations are treated as a single compound observation subject to a probability density function, but this pdf is not constrained to being the product of pdf values for individual observations. Instead, the correct hierarchical calculation is performed and used as the pdf for the entire observation sequence. The segmental HMM then moves on to considering a transition region as a segment in its own right, and so forth.

The unsatisfactory feature of the segmental HMM is that it sacrifices the economy of the inductive Baum-Welch computation within a segment. However it also sacrifices the consistency property of realised targets between segments. The correct way of handling the transition following a dwell would ensure that the realised target at the beginning of the transition was equal to the realised target of the preceding dwell. The segmental model has no way of achieving this because its mechanism for imposing consistency works only within a segment.

When we discuss vocal tract length normalisation later, we will see that certain approximations allow us to regard vocal tract length as coming from a distribution in just the same way as realised targets, and allow us to average over vocal tract lengths in such a way that all paths through the data assume the same vocal tract length, but that paths are considered for every possible vocal tract length. To get the same effect with a segmental HMM it would be necessary to treat the entire utterance as a single segment without any attempt at inductive calculation.

Meanwhile we return to the details of how a CS-HMM can be used to analyse sonorant sounds.

2.5. Choice of methods for handling transitions

During a dwell the continuous component of state is the hidden vector of realised targets $\boldsymbol{x}$. During a transition we will need to extend it to a longer vector.

One way of doing this, when we leave a phonetic unit φ, is to assume the identity of the destination φ' and the duration L of the transition. The continuous component of state can then be a $2m$-long vector $(\boldsymbol{x}, \boldsymbol{x}')$ comprising the pair of realised targets. Given L we know the expected values of observation vectors at any point during the transition as a linear function of $\boldsymbol{x}$ and $\boldsymbol{x}'$. This gives us the strongest possible model for predicting observations, but at the cost of a large branching factor.

If we proceed in this way, it is worth noting that no distribution on slopes ever needs to be written down: an implicit distribution is given by the distributions on targets and on durations, and this is all we need. But we have had to choose L from what may be a large set before seeing the observations which supply us with information about it.

We can avoid the expense of making this assumption in advance by instead extending the continuous state component into the vector $(\boldsymbol{x}, \boldsymbol{s})$ where $\boldsymbol{s}$ is the slope of $\boldsymbol{x}$. The centre of the observation distribution after h ticks will then

be $\boldsymbol{x} + h\boldsymbol{s}$. When we come to the end of the transition we can convert the assumption of a slope into an assumption of a realised target by marginalisation and bring in the distribution on realised targets as part of the calculation. The method used in Weber et al. (2014) was to do just this without making use of any prior distribution on the slope.

We no longer feel that this method can be justified. If we have spent h ticks in a dwell, then the correct formula for the alphas one step into a following transition is

$$\alpha_t(\boldsymbol{x}, \boldsymbol{s}) = (1-P(h))\, f(\boldsymbol{s})\, n(\boldsymbol{x}+\boldsymbol{s}-\boldsymbol{y}_{t-1}, e)\, \alpha_{t-1}(\boldsymbol{x}) \tag{15}$$

where $f(\boldsymbol{s})$ is the distribution on slopes. Now the term $f(\boldsymbol{s})$ has only a temporary effect since at the end of the transition we will remove it in favour of terms expressing the probabilities of the new targets and of the transition duration; and this was the reason for effectively discarding it in Weber et al. (2014). But the consequence was that when a hypothesis in a transition was compared with a hypothesis in a dwell, a term was omitted which might have influenced the result, namely the convolution of the transition alphas with $f(\boldsymbol{s})$.

So we feel that the term needs to be included and subsequently removed. Accordingly we will adopt a Gaussian prior $n(0, v)$ on slopes; v can be estimated from distributions on targets and durations.

The form of the induction will therefore be to use an m-long continuous state vector during dwells and a $2m$-long vector $(\boldsymbol{x}, \boldsymbol{s})$ during transitions. The reader should be alert to the fact that some of the terms in our notation fluctuate between being m-long and $2m$-long according to context.

At this point we are in a position to present formulae for entering a transition, but they turn out to be a special case of the formulae for stepping through a transition so that is the case we first consider.

2.6. Stepping through a transition

We assume that

$$\alpha_{t-1}(\boldsymbol{x}, \boldsymbol{s}) = k_{t-1}\, n((\boldsymbol{x}, \boldsymbol{s}) - \boldsymbol{\mu}_{t-1}, p_{t-1}) \tag{16}$$

where $\boldsymbol{\mu}_{t-1}$ is a $2m$-long mean and p_{t-1} is a $2m \times 2m$ precision. So

$$\alpha_t(\boldsymbol{x}, \boldsymbol{s}) = P'(h)\, \alpha_{t-1}(\boldsymbol{x}, \boldsymbol{s})\; n(\boldsymbol{y}_{t-1} - (\boldsymbol{x} + h\boldsymbol{s}), e) \tag{17}$$

where the first Gaussian (implicit in α_{t-1}) lies in $2m$ dimensions and the second (written explicitly) in m, making the operation of completing the square less simple than we might hope. We find that

$$\alpha_t(\boldsymbol{x}, \boldsymbol{s}) = k_t\, n((\boldsymbol{x}, \boldsymbol{s}) - \boldsymbol{\mu}_t, p_t) \tag{18}$$

where

$$p_t = p_{t-1} + \begin{pmatrix} e & he \\ he & h^2 e \end{pmatrix}, \tag{19}$$

$$\boldsymbol{\mu}_t = \{(\boldsymbol{y}_{t-1}e, h\boldsymbol{y}_{t-1}e) + \boldsymbol{\mu}_{t-1}p_{t-1}\}\, p_t^{-1}, \quad \text{and} \tag{20}$$

$$k_t = P'(h)\, \frac{k_{t-1}|p_{t-1}|^{1/2}|e|^{1/2}}{(2\pi)^{m/2}|p_t|^{1/2}} \times \exp\{-\tfrac{1}{2}[\boldsymbol{y}_{t-1}e\boldsymbol{y}_{t-1}^{\mathsf{T}} + \boldsymbol{\mu}_{t-1}p_{t-1}\boldsymbol{\mu}_{t-1}^{\mathsf{T}} - \boldsymbol{\mu}_t p_t \boldsymbol{\mu}_t^{\mathsf{T}}]\}. \tag{21}$$

(Recall that $P'(h)$ – the probability of staying in a transition – was defined in §2.2.)

2.7. Entering a transition

At each step during a dwell we take a single hypothesis at time $t-1$ and generate two hypotheses at time t. One of these, which we have already described, spends at least a further tick in the dwell; the other begins a transition to a new target. That there is only one hypothesis entering a transition is a consequence of our decision that our alphas would be functions of slope rather than of the target of the destination phonetic unit.

There is an operation we need to perform at the end of every dwell, namely multiplying k_t by $1-P(h)$. If we are in a dwell at the end of an utterance then we need to do this as part of the wrapup, and we need to do it when we move from a dwell to a transition.

Having done it we find that the formula we want to use when we enter a transition is a slight modification of the one we have already given as eq. (17) with h equal to 0. When we presented this equation previously the mean and precision of the alphas at $t-1$ were already $2m$-dimensional whereas at present they are m-dimensional. We need to extend them to the higher space by padding the vector mean with zeroes corresponding to the prior mean of the slope, and we need to replace the precision p_{t-1} by

$$\begin{pmatrix} p_{t-1} & 0 \\ 0 & v \end{pmatrix} \tag{22}$$

(v being the prior precision of slopes). When we have done this we may apply equations (18)-(21) directly.

2.8. Entering a dwell state

Entering a dwell is the most complicated part of the induction. We do it in a slightly different way than when entering a transition. We will assume that we have already computed the parameters k_t, $\boldsymbol{\mu}_t$ and p_t for hypotheses which have consumed the t observations up to and including $\boldsymbol{y}_{t-1}$. Each such hypothesis which is in a transition state will be left as it stands, ready to be extended by a further step in the same direction, and will also spawn a set of additional hypotheses in which the observation at $t-1$ is the first in a dwell. There is one

such hypothesis for every unit in the inventory, and hypotheses in this set will be extended using the method of §2.3.

In order to put a hypothesis in the right form for extending as a dwell we need to express its alphas in terms of the new realised target $\boldsymbol{x}'$ rather than in terms of the old target $\boldsymbol{x}$ and slope s.

Before doing this we have to remove the effect of the prior on slopes and bring in a timing term for ending the transition. We want to write

$$k_t\,(1-P'(h))\,\frac{n((\boldsymbol{x},\boldsymbol{s})-\boldsymbol{\mu}_t,p_t)}{n(0,v)} \tag{23}$$

as a single scaled Gaussian $\kappa n((\boldsymbol{x},\boldsymbol{s})-\boldsymbol{\gamma},g)$. We find that

$$g = p_t - \begin{pmatrix}0 & 0\\ 0 & v\end{pmatrix}, \tag{24}$$

$$\gamma = g^{-1}p_t\boldsymbol{\mu}_t, \quad \text{and} \tag{25}$$

$$\kappa = k_t\,(1-P'(h))\frac{|p_t|^{1/2}}{|g|^{1/2}|v|^{1/2}}\exp\left\{\tfrac{1}{2}\left[\boldsymbol{\mu}_t p_t \boldsymbol{\mu}_t^{\mathsf{T}} - \boldsymbol{\gamma} g \boldsymbol{\gamma}^{\mathsf{T}}\right]\right\}. \tag{26}$$

We now need to express the alphas in terms of the new realised target $\boldsymbol{x}'$ (by means of a change of variable) while marginalising out $\boldsymbol{s}$ by integration.

We shall write $\alpha'_t(\boldsymbol{x}')$ for the sum of probabilities of paths arriving at realised target $\boldsymbol{x}'$: the prime in α'_t distinguishes it from the alpha value $\alpha_t(\boldsymbol{x},\boldsymbol{s})$ we have already computed for the same time. The change of variables and marginalisation are obtained by writing

$$\alpha'_t(\boldsymbol{x}') = \int \mathrm{d}\boldsymbol{s}\,\alpha_t(\boldsymbol{x}'-h\boldsymbol{s},\boldsymbol{s}) \tag{27}$$

$$= \kappa\int \mathrm{d}\boldsymbol{s}\,n((\boldsymbol{x}'-h\boldsymbol{s},\boldsymbol{s})-\boldsymbol{\gamma},g) \tag{28}$$

$$= \kappa\,n(\boldsymbol{x}'-(\boldsymbol{\gamma}_{\boldsymbol{x}}+\boldsymbol{\gamma}_{\boldsymbol{s}}r_{\boldsymbol{sx}}r_{\boldsymbol{xx}}^{-1}),r_{\boldsymbol{xx}}) \tag{29}$$

which is in the required form, where

$$r = g - g\begin{pmatrix}h^2q^{-1} & -q^{-1}\\ -hq^{-1} & q^{-1}\end{pmatrix}g \quad \text{and} \tag{30}$$

$$q = h^2 g_{\boldsymbol{xx}} - h(g_{\boldsymbol{xs}}+g_{\boldsymbol{sx}}) + g_{\boldsymbol{ss}} \tag{31}$$

and where vector subscripts denote quadrants of matrices (so that $r_{\boldsymbol{xx}}$ is the first, ie. $(\boldsymbol{x},\boldsymbol{x})$, quadrant of r). This calculation gives us the sum of path probabilities over all paths leading to a putative new target $\boldsymbol{x}$. The path probabilities are products of observation probabilities over all observations seen so far together with realisation probabilities for all targets along the path (excluding the target we are now reaching) and associated information from language and timing models. To proceed further we need to bring in the missing terms relating to the new target by assuming its phonetic identity and multiplying the

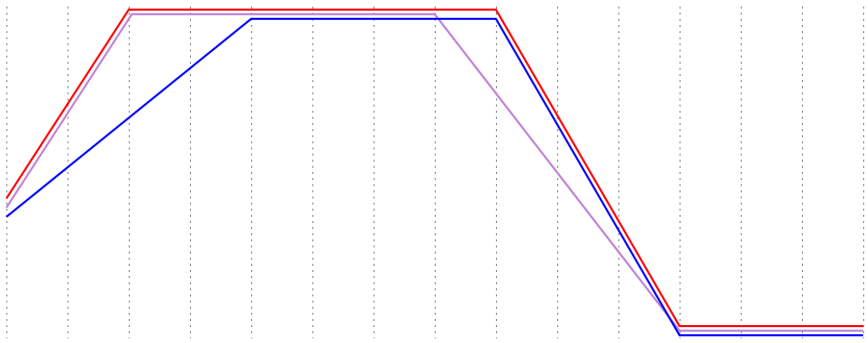

Figure 3: Three formant hypotheses which agree phonetically but disagree in their timing. We have a choice between choosing the best of them or combining them into a single hypothesis with the sum of their probabilities.

alphas by $n(\boldsymbol{x} - \boldsymbol{\theta}_{\boldsymbol{\varphi}}, c_{\varphi})$ and by a term from the language model. They are then in the form we assumed earlier when we stepped through a dwell, allowing us to continue the induction.

There is another question to consider when we enter a dwell. As we described the algorithm at the outset, a hypothesis at any given time derives from a unique antecedent at any earlier time. If we adopt this principle, then we will ultimately recover the likeliest sequence of discrete states. This amounts to recovering the likeliest phonetic sequence together with its assumed timings. Figure 3 illustrates some hypotheses which are equivalent phonetically but differ in timing.

For most purposes it is more useful to recover the likeliest sequence of phonetic units averaged over all consistent timings. In this case, when two hypotheses with the same phonetic history enter a dwell state with the same assumed phonetic unit, we should combine them into a single hypothesis. The composite hypothesis will need the sum of the alpha distributions of its constituents; and since the sum of scaled Gaussians is not itself a scaled Gaussian we will need to perform an approximation. (We will hope that the means and variances of the scaled Gaussians are similar with the consequence that we are effectively adding the scale factors.)

2.9. Transient dwells

The formulae presented so far allow trajectories to be followed through the entire utterance. Dwells of length zero need no special treatment. The method of §2.8 allows us to convert the final step of a transition into the first step of a dwell, and if on the next tick we apply the method of §2.7 we will have limited the dwell to length 0.

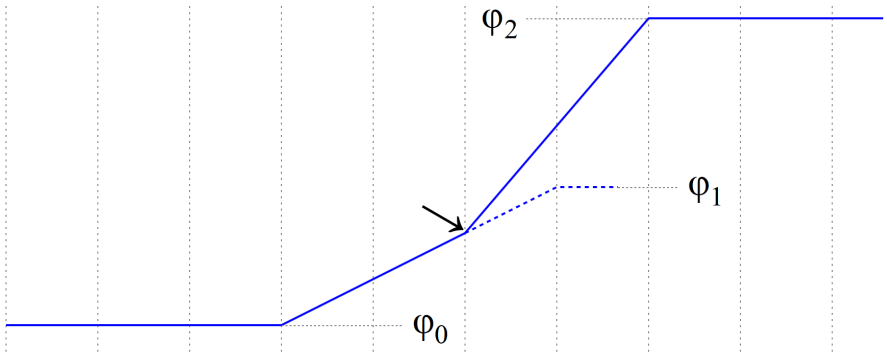

Figure 4: A formant commences a transition from φ_0 to its successor φ_1 but changes direction towards φ_2 before reaching it. We view φ_1 as having a dwell of length -1.

2.10. Negative dwells

We use the term *negative dwell* to denote a phenomenon which arises in rapid speech. The articulators embark on a transition from one phonetic unit φ_0 to its successor φ_1, and before reaching their destination veer off in the direction of the following unit φ_2 as shown in Figure 4. Other names for the same phenomenon are *undershoot* and *overlapping transitions*.

Negative dwells can be handled with little change to the method described earlier. At the point in which the trajectory changes direction from heading to φ_1 to heading to φ_2 we will have alphas written as $\alpha_t(\boldsymbol{x}, \boldsymbol{s})$. We will rewrite them as $\alpha'_t(\boldsymbol{x}')$ by applying a similar method to that of §2.8. $\boldsymbol{x}'$ will be the realised target interpolated between φ_1 and φ_2. We will not subsequently be able to bring in a term like $n(\boldsymbol{x} - \boldsymbol{\theta}_{\boldsymbol{\varphi}}, c_\varphi)$ reflecting the match of the realised to the canonical target of φ_1 because we never adopt the realised target as a continuous state component. Instead this term has to be brought in during the marginalisation. So the first step is to multiply by $1{-}P'(h)$ and divide by $n(0, v)$ as in equations (23)-(26). We then perform a marginalisation in which equation (27) is replaced by

$$\alpha'_t(\boldsymbol{x}') = \int \mathrm{d}\boldsymbol{s}\, \alpha_t(\boldsymbol{x}' - h\boldsymbol{s}, \boldsymbol{s})\; n(\boldsymbol{x}' + h's - \boldsymbol{\theta}_{\boldsymbol{\varphi}}, c_\varphi) \tag{32}$$

where h' is the number of ticks we are short of reaching the φ_1 dwell (so we would describe this case as having a dwell duration of $-h'$).

We also need to multiply the scale factor by the timing probability of the dwell length $-h'$.

Once we have alphas expressed in terms of the realised continuous component at this intermediate point, we can pad out the vectors to length $2m$ using the prior mean 0 and precision v of the new slope and use the method of §2.6 to step through the second transition.

We notice that if a wide range of negative dwell times is permitted, the number of options to consider at each point in a transition becomes large. We

also remark that if negative dwells are allowed by the model, then the $P(-1)$ of equations (7)-(10) needs to be redefined as the probability that a dwell-length will be non-negative.

3. Experiments on synthetic sinusoidal speech

3.1. Summary of the experiments

Preliminary results from applying a CS-HMM to real speech from the TIMIT corpus are reported in Weber et al. (2014); but here we describe results on toy data which validate the algorithm under a widely accepted model of sonorant sounds. We have made improvements in our results for real speech since our earlier paper was published, but we are not at the point that the results cast light on the soundness of our approach rather than on the completeness of our implementation. This remains work in progress.

In our experiments sinusoidal speech is synthesised from a randomly generated phonetic inventory using a simplified HMS model. It comprises 'formants' following HMS trajectories, but these pseudoformants are sine waves during dwells and swept sine waves during transitions. Three formants are used for the experiment.

The realised targets are generated from a Gaussian distribution centred on the canonical targets with standard deviation σ_f Hz. Dwell and transition durations are taken from uniform distributions: transitions range from 2 to 6 frames (inclusively) whereas dwells range from 0 to 4 frames in one experiment and from 1 to 4 in the others.

Canonically all formants have the same amplitude, but realised amplitudes are taken from a lognormal distribution with parametric variance: we will see what effect this has on the recovery process. In the first two experiments the standard deviation of log amplitudes is 0.3; in the third it is 0. Transitions move linearly between the frequencies and amplitudes of their end points.

The inventory comprises 40 units whose canonical targets are chosen from a uniform distribution over 200-3800Hz subject to the constraint that no pair of adjacent formants has canonical targets closer than 150Hz (and that the frequencies are in ascending order). The language model is uniform except that a phonetic unit can never be followed by itself (so the legal transitions each have probability 1/39).

3.2. Algorithms considered

We evaluate the CS-HMM of this paper against the standard DS-HMM, running each algorithm with two different feature sets. The model used by the CS-HMM analyses the data into dwells and transitions as described previously. The DS-HMM uses a conventional model in which each phonetic unit is divided into 3 substates, the first being the second half of the transition from the predecessor, the second being the dwell, and the third being the first half of the transition to the successor. Transition substates are modelled separately for every pair of surrounding phonetic units, but dwells are modelled independently of context. This reflects the intended interpretation of the substates.

The first feature set (denoted '(f)') comprises simulated noisy formant frequency measurements, obtained from the true pseudoformant frequencies subject to Gaussian measurement error which has a parametric standard deviation. For the DS-HMM we append Δ-frequencies (which we didn't find to be very useful).

The second feature set, denoted '(a)', is derived directly from the acoustics, which is done differently for the two algorithms with the noise in each case being additive Gaussian acoustic white noise with parametric SNR.

For CS-HMM (a) we converted the acoustic input into formant frequency estimates using a crude formant tracker, made up of a Fourier tone detector which analyses spectral peaks as tones for each frame independently and is followed by a dynamic program which imposes continuity on the frequencies of the peaks selected. We make no attempt to model transitions as swept tones rather than as tones, or to resolve broad peaks into multiple tones, or to exploit amplitude continuity in the dynamic program, or to take advantage of phase-coherence between frames (which would be cheating, given that no such property could be exploited in speech).

For DS-HMM (a) we extracted 13-long cepstral vectors and experimented with adding Δ-cepstra and Δ^2-cepstra. We found that Δ-cepstra were always beneficial, and that Δ^2s were beneficial for the higher SNRs (20dB and 60dB) but harmful when the SNRs were lower (ie. for noisier signals). We quote results for whichever of the two worked better.

All the algorithms were trained on a further 4 hours of data synthesised using the parameters of the test set. We made the training easier by using the known phonetic labels rather than by using Baum-Welch or Viterbi alignment.

The number of parameters differs between algorithms. Both the CS-HMMs have 40 vector means, 40 realisation variance matrices, and a single observation variance matrix; all of which apply to 3-long features. In our experiments the realisation variances were pooled and diagonal but we don't expect these properties to hold for real signals. The observation variance is pooled between phonetic units and is a scalar multiple of the identity.

The DS-HMMs each have 40 mean and variance matrices for the central substate of each phonetic unit, together with a further $2\times40\times39$ for the substates assigned to transitions. The variance matrices are assumed diagonal. The parameter space differs between DS-HMM (f) and DS-HMM (a) only in the length of their feature vectors: 6 in the first case, 26 or 39 in the second.

3.3. Experimental rationale

The experiment was designed with the aim of validating the effectiveness of the decoding algorithms (and hence their associated models) without confusing the results with uncertainty over whether the algorithms had been adequately trained. This was achieved by using a generous training corpus with oracular labels (there was no need for any form of EM training).

We gave the DS-HMM no more parameters than could be seen to serve a purpose. There is no real justification for using GMMs in preference to pure

Gaussians given the nature of the generative process. This would have expanded the parameter space and would have necessitated EM training which might have cast doubt on the results. The phonetic context is one-sided and limited to a single unit since no additional context has any influence on the generation process.

We have experimented informally with Viterbi training of the CS-HMM on the same data and found that it worked perfectly well.

3.4. First experiment: variable amplitudes, dwell times from 0 to 4

We now present results of the first experiment (Table 1). Each cell contains the error rate (computed using SCLite (Fiscus et al., 2006)) for a particular configuration. The error rate is averaged over 20 experiments, each with a randomly generated phonetic inventory giving rise to an audio stream comprising 1000 phonetic units.

The four blocks across the table contain the results for the two algorithms and their differing feature sets. There are 3 columns in each block corresponding to different variabilities of realised pseudoformant target frequencies about their canonical mean, quoted as standard deviations σ_f.

The top row corresponds to the almost complete absence of noise. It is legitimate to compare the accuracy of all four blocks here. In subsequent rows the level of noise increases but on different scales depending on the feature set, and is either the standard deviation σ_n of the synthetic observations about the true value or the SNR for acoustic data. For this reason it is not legitimate to compare results across the double vertical bar.

In each block the easiest problems lie at top left and the hardest at bottom right.

Comparing the first two blocks we see that the CS-HMM is more effective than the DS-HMM in capturing the dynamic properties of its input in spite of having a much smaller parameter space.

In the right-hand half of the table we see that DS-HMM (a) is somewhat weaker than DS-HMM (f) in the top row but stronger than CS-HMM (a) throughout. The poor performance of CS-HMM (a) is nothing to do with any weakness in the decoding algorithm: the difficulty is in tracking the pseudoformants. When a quiet F2 makes a rapid transition from close to F1 to close to F3 and back, the swept tone is almost invisible to the tone detector and the dynamic program prefers a hypothesis in which F2 stays close to F1. We expect that these effects could be obviated if we were willing to write a sufficiently sophisticated pseudoformant tracker, but that is too large a task to be justified for artificial data.

It is worth noting that CS-HMM (a) comes closest to DS-HMM (a) when σ_f is large. We assume that this is because the CS-HMM then gains a larger advantage through modelling the two forms of variability separately.

3.5. Second experiment, excluding zero dwells

This second experiment allows dwells to range from 1 to 4 frames, with results as shown in Table 2.

	CS-HMM (f)			DS-HMM (f)			CS-HMM (a)			DS-HMM (a)			
σ_f	10	30	60	10	30	60	10	30	60	10	30	60	
$\sigma_n = 1$	0.06	0.72	3.59	3.74	5.54	10.34	22.78	23.06	26.42	6.84	9.97	19.48	60 = SNR
10	0.06	0.68	3.68	3.62	5.53	10.71	22.26	22.92	26.46	8.42	11.92	23.03	20
30	0.24	0.88	3.86	4.69	6.31	11.47	21.26	21.86	26.32	9.82	13.62	25.13	10
60	1.00	1.92	5.14	7.33	8.85	13.58	31.00	31.98	34.74	19.81	23.33	33.22	0

Table 1: Phonetic error rates (as percentages) when dwells are in the range $[0, 4]$ and formant amplitudes are variable (their logs have standard deviation 0.3). The leftmost column gives σ_n, the measurement noise for formant features, whereas the rightmost gives the SNR in dB determining the additive noise for acoustic features. The three columns per algorithm correspond to different values of σ_f, the standard deviation of realised frequencies about their canonical means.

	CS-HMM (f)			DS-HMM (f)			CS-HMM (a)			DS-HMM (a)			
σ_f	10	30	60	10	30	60	10	30	60	10	30	60	
$\sigma_n = 1$	0.03	1.00	3.69	0.14	1.26	5.56	16.57	18.22	21.86	4.55	7.18	15.46	60 = SNR
10	0.04	0.46	3.62	0.20	1.24	5.58	16.70	18.14	21.50	5.22	8.68	17.55	20
30	0.17	0.87	3.62	0.71	1.88	6.12	16.46	17.73	21.42	6.78	9.66	19.03	10
60	0.74	1.53	4.62	2.80	3.74	7.86	23.94	25.94	29.06	15.09	18.04	28.01	0

Table 2: Error rates as in Table 1 but with zero-length dwells excluded. Here, dwell times are in the range $[1, 4]$.

	CS-HMM (f)			DS-HMM (f)			CS-HMM (a)			DS-HMM (a)			
σ_f	10	30	60	10	30	60	10	30	60	10	30	60	
$\sigma_n = 1$	0.03	1.00	3.69	0.14	1.26	5.56	1.73	3.10	6.78	4.97	7.89	16.30	60 = SNR
10	0.04	0.46	3.62	0.20	1.24	5.58	1.82	3.02	6.69	4.86	7.82	17.05	20
30	0.17	0.87	3.62	0.71	1.88	6.12	1.68	2.84	6.70	5.99	8.66	18.66	10
60	0.74	1.53	4.62	2.80	3.74	7.86	4.06	5.04	9.38	10.48	13.44	23.12	0

Table 3: Error rates as in Table 1. Dwell times are in the range $[1, 4]$ and formant amplitudes are fixed. Each of these changes makes the problem a little easier.

We see that DS-HMM (f) gets closer to CS-HMM (f) than in the previous experiment, which is what we would expect given that its weakness is in handling dynamics and that we have increased the proportion of information available from the static components. The gap is still appreciable.

3.6. Third experiment with constant pseudoformant amplitudes

Finally we remove the variability of pseudoformant amplitudes. Results are displayed in table 3.

This experiment makes the task of the peak picker a good deal easier by removing the risk that pseudoformants will be too quiet to be detected when moving quickly. As a result CS-HMM (a) outperforms DS-HMM (a) by a significant margin. (The results with formant features are the same as in the previous table because they are independent of amplitude.)

4. Extensions of the model

4.1. Vocal tract length

So far we have described a speaker-independent algorithm for recovering a phonetic sequence given acoustic measurements. We now look at extensions of the model which take explicit account of the acoustic and phonetic phenomema of vocal track length (VTL), of the loudness of acoustic elements, and of channel effects. In order to consider these topics we need to make concrete assumptions about the nature of the feature vectors.

The first extension to consider introduces a degree of speaker adaptation by treating vocal tract length as an additional component of the hidden continuous state. The main effect of vocal tract length is to apply a scalar multiplier to formant frequencies or to articulator positions. But in order to incorporate VTL into our model we need the feature vector to have the logs of these quantities as components rather than linear values. So assume, for the following discussion, that the feature vector comprises precisely the logs of m measured formant frequencies.

This assumption has certain effects on our model. It implies an expectation that formant trajectories will not be linear after all, but will follow the exponential curves implied by linearity of the logs. We do not feel that this makes the model any less plausible since there are no convincing reasons to prefer one trajectory shape to the other.

We may specify the canonical phonetic targets for a 'standard' speaker. We add an extra scalar λ to the continuous state, which is the log ratio of the speaker's vocal tract length to the standard: λ comes from a 0-mean Gaussian distribution. We then view the realised targets as varying about a mean of the form $\boldsymbol{\theta}_\varphi + \lambda \mathbf{1}_m$ where $\boldsymbol{\theta}_\varphi$ is (componentwise) the logarithm of the mean canonical formant frequencies for a standard speaker and $\mathbf{1}_m$ is an m-long vector of 1s.

We apply the CS-HMM in exactly the same way as before. v is assumed to be constant throughout the data. At the end we have a preferred recovery of the phonetic sequence and an alpha value which takes the form of a Gaussian

distribution on an $(m+1)$-dimensional vector comprising m log formant targets and λ. Projecting onto a single dimension gives us the posterior distribution of the speaker's vocal tract length, which may be useful for some purposes.

The precision matrices used by the CS-HMM now specify variability in the logs of formant frequencies rather than in their linear values. There is no reason to suppose that this will give an appreciably worse fit to the variability of realised about canonical targets than we would obtain using linear formant frequencies.

The biggest drawback to working in the log space comes when we look at the precision matrix e reflecting variability of formant measurements about their true values. In linear formant space we expect this to be a scalar multiple of the identity matrix, although nothing in the analysis requires it to take this form. In a log space it is impossible to provide an entirely satisfactory substitute. The reason is that we expect errors to be of roughly equal magnitude throughout the frequency range, which implies that we expect errors in log frequencies to be greater when the true frequencies are low than when they are high. We can allow for this to a certain extent by stipulating greater variance for log F1 than for log F3. However F1 itself varies by a factor of more than 5 between its extrema, so this compensation is insufficient.

For any given formant frequency, we expect observations to come from a Gaussian distribution centred on the true frequency. There is no difficulty in finding a log-normal distribution which closely matches the given Gaussian: the trouble is that identifying the right log-normal distribution depends on knowing the true frequency. But although the true frequency is unknown, it is likely that the measured frequency will be close to it, so at any point we can find a log-normal distribution which is likely to be approximately correct. (Unsmoothed formant measurements are permissible as inputs to the CS-HMM, but we should smooth them for the purpose of fitting a log-normal to a Gaussian distribution.)

This is certainly not as rigorous or as exact as the methods of earlier in this paper, but nor is there anything illegitimate about what we have done. Observations and system parameters are both assumed known at the outset, and we proceed to recover and marginalise the various unknowns. There's nothing to prevent us from looking at one set of knowns – the observations – before settling on the other – the system parameters.

An alternative method of handling VTL, which is almost the only way available to conventional models, is to perform a speaker-independent trial decode, to estimate speaker characteristics from it, and to decode afresh using an improved model. This is clumsy compared with "tuning in" in the way we have described.

4.2. Loudness and channel effects

If the acoustic features are made up of formant frequencies then loudness has no effect on recognition. (By loudness, without being precise, we mean some linear transformation of log amplitude.) However it is unlikely that formant frequencies alone contain all the information needed.

If the acoustic features are spectral band energies, then loudness and channel effects can be brought in in much the same way as VTL can be for formant frequencies.

A third option which merits consideration is for the acoustic features to contain components analogous to formant loudnesses in addition to features derived from formant frequencies. However we do not want recognition results to be influenced by the overall loudness of the signal, which is not phonetically relevant. So the first step is to introduce the overall loudness level as another persistent scalar state variable in the same way as log VTL previously, and to initialise it with a prior with low precision. The relative loudness of certain components (for instance the quietness of F3 for nasal consonants) can now be exploited during recognition.

If we wish to allow for unknown linear channel effects we need to do more than this. These effects lead to a frequency-dependent amplification of the signal. We could represent it in the model by postulating a number of persistent state variables corresponding (say) to the channel effects at 0Hz, 1000Hz, 2000Hz and 3000Hz, understanding effects at intermediate frequencies to be given by linear interpolation. The prior assumption will be that neighbouring channel effects take similar values. If we then take the estimated formant frequency at any point as a proxy for the true frequency, we can infer an expected loudness as a linear function of state variables, and compare it with the measured loudness under a Gaussian density function.

It is interesting to note that this procedure is neater than the techniques available in the cepstral domain, even though channel effects are simple linear additives there. But they can be expressed adequately in frequency space, and the CS-HMM then allows them to be modelled as unseen state components in a way which a DS-HMM cannot achieve.

5. Conclusions

In this paper we have outlined a model for sonorant speech based on that of Holmes, Mattingly and Shearme and we have developed an optimal algorithm for extracting information from the dynamics implied by it. We have validated the algorithm in experiments on toy data and we have shown how it can be extended to capture other important effects. Along the way we have shown that the dynamical structure of speech is imperfectly preserved by direct use of cepstral measurements.

Questions for further consideration are:

- How accurately can sonorant speech be modelled by piecewise linear formant frequencies and loudnesses?
- What should we do about non-sonorant sounds (such as unvoiced consonants)?
- How can we measure the acoustic features we need?

- Or if we cannot measure them reliably, can we modify the algorithm to incorporate them as a hidden layer?

6. Acknowledgements

The authors would like to thank Martin Russell, Peter Jančovič and Phil Weber for their help in writing this paper; and the anonymous referees for advice which added much to its quality.

7. References

Ainsleigh, P.L., 2001. Theory of continuous-state hidden Markov models and hidden Gauss-Markov models. Technical Report NUWC-NPT Technical Report 11274. Naval Undersea Warfare Center Division (Newport, Rhode Island).

Baum, L.E., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics 41 (1), 164–171.

Bellman, R., 1954. The theory of dynamic programming. Bulletin of the American Mathematical Society 60 (6), 503–515.

Deng, L., 1998. A dynamic, feature-based approach to the interface between phonology and phonetics for speech modelling and recognition. Speech Communication 24 (4), 299–323.

Deng, L., Cui, X., Pruvenok, R., Huang, J., Momen, S., Chen, Y., Alwan, A., 2006. A database of vocal tract resonance trajectories for research in speech processing, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 369–372.

Deng, L., Ma, J., 2000. Spontaneous speech recognition using a statistical coarticulatory model for the vocal tract resonance dynamics. Journal of the Acoustical Society of America 108 (6), 3036–3048.

Fiscus, J.G., Radde, N., Ajot, J., Laprun, C., 2006. Multiple dimension Levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech, in: Proceedings of the Fifth International Conference on Language Resources and Evaluation, pp. 803–808.

Furui, S., 1981. Comparison of speaker recognition methods using statistical features and dynamic features. IEEE Transactions on Acoustics, Speech and Signal Processing 29 (3), 342–350.

Gales, M.J.F., Young, S.J., 1993. Segmental Hidden Markov Models, in: Proceedings of Eurospeech, pp. 1579–1582.

Gales, M.J.F., Young, S.J., 2007. The application of Hidden Markov Models in Speech Recognition. Foundations and Trends in Signal Processing 1 (3), 195–304.

Holmes, J., Mattingly, I., Shearme, J., 1964. Speech synthesis by rule. Language and Speech 7, 127–143.

Holmes, J.N., 2001. Speech processing system using formant analysis. US patent US6292775 .

Holmes, J.N., Holmes, W.J., 2001. Speech synthesis and recognition. 2nd ed., Taylor & Francis.

Holmes, W.J., Russell, M.J., 1999. Probabilistic-trajectory segmental HMMs. Computer Speech and Language 13 (1), 3–37.

Houghton, S.M., 2014. CS-HMM source code: `https://bitbucket.org/uobsrbs/cshmm-public`.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82 (1), 35–45.

King, S., Frankel, J., Livescu, K., McDermott, E., Richmond, K., Wester, M., 2007. Speech production knowledge in automatic speech recognition. Journal of the Acoustical Society of America 121 (2), 723–742.

Paliwal, K.K., Rao, P.V.S., 1982. Synthesis-based recognition of continuous speech. Journal of the Acoustical Society of America 71 (4), 1016–1024.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., 2011. The kaldi speech recognition toolkit, in: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Richards, H.B., Bridle, J.S., 1999. The HDM: a segmental hidden dynamic model of coarticulation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 357–360.

Richmond, K., King, S., Taylor, P., 2003. Modelling the uncertainty in recovering articulation from acoustics. Computer Speech and Language 17, 153–172.

Russell, M.J., 1993. A segmental hmm for speech pattern matching, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 499–502.

Russell, M.J., Jackson, P.J.B., 2005. A multiple-level linear/linear segmental HMM with a formant-based intermediate layer. Computer Speech and Language 19 (2), 205–225.

Russell, M.J., Zheng, X., Jackson, P.J.B., 2007. Modelling speech signals using formant frequencies as an intermediate representation. IET Signal Processing 1 (1), 43–50.

Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Proceedings of Interspeech, pp. 338–342.

Tokuda, K., Zen, H., Kitamura, T., 2003. Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features, in: Proceedings of Eurospeech, pp. 865–868.

Weber, P., Houghton, S.M., Champion, C.J., Russell, M.J., Jančovič, P., 2014. Trajectory analysis of speech using continuous state Hidden Markov Models, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3042–3046.

8. Vitae

Colin Champion has been working in speech since 1997. He is currently an Honorary Research Fellow of the University of Birmingham. His main interests are Bayesian (and especially empirical Bayesian) statistics; DSP theory; Hidden Markov Model theory; and the acoustic analysis of speech.

Steve Houghton received his B.A, M.Math and Ph.D. degrees from the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK in 2000, 2001 and 2006 respectively. After spending time as a researcher in the areas of nonlinear dynamics, pattern formation and astrophysical fluid dynamics, he began working in the area of speech recognition in 2011 and is an Honorary Research Fellow at the University of Birmingham, UK. He is a member of IEEE, ISCA, SIAM and a Fellow of the Royal Astronomical Society.