Linear dynamic models for automatic speech recognition

Joe Frankel

Thesis submitted for the degree of Doctor of Philosophy

University of Edinburgh

April 2003

Dedication

To Eric, the original Dr. Frankel.

Acknowledgements

Firstly, utmost thanks to Simon King for his generosity. This thesis would never have reached fruition without his insight and guidance, and I count myself very lucky to have had someone with such an inspirational teaching style assume the role of my supervisor. Thanks also to all those who make the CSTR such a stimulating and supportive environment to work in. Those deserving of a special mention include Korin Richmond for his attention to detail and willingness to discuss/share knowledge of matters ranging from computing to linguistics, through machine learning, onto pottery, politics and home-brew. Also Rob Clark for always making time to sort out my linux and system-related hiccups despite having plenty of his own work to do. Thanks to those who read all or parts of this thesis while it was in preparation – your comments and questions have been invaluable.

Thanks to those who have given me useful feedback over the last years, or taken the trouble to answer queries which arose at odd times. I've benefited from the input of Paul Taylor, Steve Isard, Amos Storkey, Chris Williams, Sam Roweis, Zoubin Gharamani and Gavin Smith.

Thanks to both my English and Norwegian families, whose on-going support of every kind has made it possible to see this thesis through to completion. Also to those who make Edinburgh and Woodcote life such a pleasure – especially the inhabitants of [the flat known as] 2f2 who have so graciously tolerated my living in the country, camping in town, and turning up after late nights at the office for a glass of IPA in their kitchen.

So finally to Synnøve, who has been an amazing companion over the last years...shared with me so many good times, survived the fallout from the emotional roller-coaster that is a PhD, and produced the lovely Eva, who I hope will not remember how busy her father was during her first six months.

Declaration

I have composed this thesis. Unless otherwise stated, the work reported is my own.

Abstract

The majority of automatic speech recognition (ASR) systems rely on hidden Markov models (HMM), in which the output distribution associated with each state is modelled by a mixture of diagonal covariance Gaussians. Dynamic information is typically included by appending time-derivatives to feature vectors. This approach, whilst successful, makes the false assumption of framewise independence of the augmented feature vectors and ignores the spatial correlations in the parametrised speech signal. This dissertation seeks to address these shortcomings by exploring acoustic modelling for ASR with an application of a form of state-space model, the linear dynamic model (LDM).

Rather than modelling individual frames of data, LDMs characterise entire segments of speech. An auto-regressive state evolution through a continuous space gives a Markovian model of the underlying dynamics, and spatial correlations between feature dimensions are absorbed into the structure of the observation process. LDMs have been applied to speech recognition before, however a smoothed Gauss-Markov form was used which ignored the potential for subspace modelling. The continuous dynamical state means that information is passed along the length of each segment. Furthermore, if the state is allowed to be continuous across segment boundaries, long range dependencies are built into the system and the assumption of independence of successive segments is loosened. The state provides an explicit model of temporal correlation which sets this approach apart from frame-based and some segment-based models where the ordering of the data is unimportant. The benefits of such a model are examined both within and between segments.

LDMs are well suited to modelling smoothly varying, continuous, yet noisy trajectories such as found in measured articulatory data. Using speaker-dependent data from the MOCHA corpus, the performance of systems which model acoustic, articulatory, and combined acoustic-articulatory features are compared. As well as measured articulatory parameters, experiments use the output of neural networks trained to perform an articulatory inversion mapping. The speaker-independent TIMIT corpus provides the basis for larger scale acoustic-only experiments. Classification tasks provide an ideal means to compare modelling choices without the confounding influence of recognition search errors, and

are used to explore issues such as choice of state dimension, front-end acoustic parametrisation and parameter initialisation. Recognition for segment models is typically more computationally expensive than for frame-based models. Unlike frame-level models, it is not always possible to share likelihood calculations for observation sequences which occur within hypothesised segments that have different start and end times. Furthermore, the Viterbi criterion is not necessarily applicable at the frame level. This work introduces a novel approach to decoding for segment models in the form of a stack decoder with A^* search. Such a scheme allows flexibility in the choice of acoustic and language models since the Viterbi criterion is not integral to the search, and hypothesis generation is independent of the particular language model. Furthermore, the time-asynchronous ordering of the search means that only likely paths are extended, and so a minimum number of models are evaluated.

The decoder is used to give full recognition results for feature-sets derived from the MOCHA and TIMIT corpora. Conventional train/test divisions and choice of language model are used so that results can be directly compared to those in other studies. The decoder is also used to implement Viterbi training, in which model parameters are alternately updated and then used to re-align the training data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

In Bourlard, Hermansky & Morgan (1996), a trio of prominent speech scientists confronted the speech recognition research community. With the belief that current techniques would not ultimately provide the performance sought from speech recognizers, they urged researchers to look beyond incremental modification of standard approaches. They acknowledged that this would lead to increases in word error rate in the short-term, but that this should not act as discouragement if three criteria are met. These are:

- solid theoretical or empirical motivations
- sound methodology
- deep understanding of state-of-the-art systems and of the specificity of the new approach.

With these in mind, it is time to start building the case for the departure from the mainstream of automatic speech recognition with which this thesis is concerned.

1.2 Speech recognition today

Automatic speech recognition (ASR) has moved from science-fiction fantasy to daily reality for citizens of technological societies. Some people seek it out, preferring dictating to typing, or benefiting from voice control of aids such as wheel-chairs. Others find it embedded in their hi-tec gadgetry – in mobile phones and car navigation systems, or cropping up in what would have until recently been human roles such as telephone booking of cinema tickets. Wherever you may meet it, computer speech recognition is here, and it's here to stay.

Statistical methods have come to dominate ASR. Speech data provides a huge amount of information to build a model upon, data into which variability is introduced by diverse factors such as the speaker's dialect, vocabulary, mood, rate of speech, breathiness, type of microphone and presence of background noise. Used properly though, all of these are information sources which can be combined, along with their associated uncertainties, by statistical models.

1.2.1 Formulation of the problem

A typical ASR system has four main components:

- **front-end parameterization** which derives an N-length sequence of p-dimensional features $\mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ from the acoustic signal. Current techniques assume that spectral characteristics are constant over the interval of a few milliseconds. The speech signal is windowed into overlapping regions of fixed duration each of which gives rise to one 'frame' of observations $\mathbf{y}_t$.

- **acoustic modelling** in which each model of a set $\mathcal{M}$ is trained to characterise a distinct unit of sound. Context dependent phone units are common. The acoustic model's function during recognition is to give $p(\mathbf{y}_1^N|m_1^k)$[1], the likelihood that an observation sequence $\mathbf{y}_1^N$ was generated by the model sequence $m_1^k = \{m_1, \ldots, m_k\}$. These models can take a variety of forms, though frame-based hidden Markov models (HMM) and artificial neural networks are most commonly used. These are described in Sections 1.3.1 and 1.3.2 below. An alternative approach is that of segment modelling as described in Section 1.3.3, which aims to characterise entire units of speech.

[1]The usual notations are followed with $P(\cdot)$ and $p(\cdot)$ denoting probability mass and density respectively.

- **language modelling** which assigns a prior probability $P(w_1^j)$ to any candidate word string $w_1^j = \{w_1, \ldots, w_j\}$. A lexicon maps each word in the dictionary onto the possible sequences of sub-word models from which it can be formed. The lexical probability is written $P(m_1^k|w_1^j)$ and is simply unity unless the lexicon allows multiple pronunciations.

- **decoding** in which a search is made for the most likely sequence of words $\hat{w}_1^j$ given the observations $\mathbf{y}_1^N$, using the acoustic, lexical and language model probabilities:

$$\hat{w}_1^j = \underset{w_1^j}{\operatorname{argmax}}\, p(\mathbf{y}_1^N|m_1^k)\, P(m_1^k|w_1^j)\, P(w_1^j) \tag{1.1}$$

The main concern of this thesis is acoustic modelling, though of course meaningful testing of new acoustic models also requires the other components. The signal processing, language modelling and decoding needed for the classification and recognition experiments in this thesis are described in Sections 3.3, 3.1.2 and 6.2 respectively.

1.3 Acoustic modelling for ASR

An observation by Jordan Cohen is paraphrased by Bourlard et al. (1996):

> ...[there is a] distinction between speech *description* (which we usually do by increasing the number of models and parameters) and speech *modelling* (which is the goal of extracting the underlying properties and invariants of the speech signal).

Though the current generation of ASR systems rely strongly on speech description, performance does continue to improve. This is almost entirely due however to larger training corpora and faster processors which can support ever larger unit inventories and numbers of model parameters. Many of those working with speech technology believe that speech *modelling* will be the route into real improvements in ASR performance. Roweis (1999) makes the case for speech-production motivated ASR by drawing comparisons with other engineering problems. Typically in cases where inference is sought from some noisy observations, the first step is to devise an appropriate model. Such a model will usually have internal states which reflect the nature of the process. The model is trained on the available data, and then used to make inferences about new and unseen observations. If this kind of approach is to be applied to ASR, the properties of such a model must first be considered. The ideal acoustic model for ASR would:

- reflect the production mechanism – i.e. be constrained, continuous and dynamic
- model contextual effects such as co-articulation in a principled manner
- make distinct models of each separate source of variation/noise, such as phonological or speaker variation, background noise
- account for the temporal dependencies in speech data
- model the spatial correlations between dimensions of the acoustic features

and on a practical level, there must be

- some means of training the model given appropriate data
- a method of comparing models, i.e. evaluating $p(\mathbf{y}_1^N|m_1^k)$
- sufficiently low computational cost to allow real-time recognition

Hidden Markov model (HMM) and hybrid artificial neural network (ANN) /HMM systems provide the basis for the majority of ASR applications, as it is these approaches which have provided the necessary combination of accuracy and efficiency. The last 15 years have also seen steady and continued interest in statistical segment modelling from a variety of groups and researchers. However, despite the theoretical advantages of this class of models, none have made it into mainstream ASR. The following sections discuss these three different approaches with reference to the acoustic modelling ideals above.

1.3.1 Hidden Markov Models

An HMM $m \in \mathcal{M}$ consists of a number of discrete states $q_j \in \mathcal{Q}_m$, and associated with each is a Gaussian mixture model over the observations. Each state aims to characterise a distinct, stationary region of the acoustic signal. Phone or word models are then formed by concatenating the appropriate set of states. The *hidden* part of the HMM is the state process, which is governed by a Markov model. Since distributions rather than symbols are associated with each state, there is no unique mapping between state and observation, and it is the goal of recognition to determine the most likely underlying state sequence[2]. Training HMMs uses an application of the expectation maximisation (EM) algorithm which is described in Section 2.1.1 on page 16. For each state, an output distribution and set of transition probabilities must be estimated.

The HMM is a frame-based model. When conditioned on a discrete state $q_j \in \mathcal{Q}$, successive frames of data are assumed to be independent. This has the effect of ignoring the temporal ordering of the observations within each state. To add the capacity to model inter-frame dependencies, dynamic information is often included in the form of δ and $\delta\delta$ coefficients. These can be simple differences, though are more often computed using some form of numerical differentiation. For instance, HTK (Young 1993), a widely used HMM recognition toolkit, estimates derivative information at a given time using a linear regression over the previous and following 2 frames (Young 1995). Therefore, contained in the $\delta\delta$ coefficients is information from the surrounding 8 frames. Including time-derivatives takes account of the ordering of the original features and yields significant improvements in recognition performance, though makes the assumption of framewise independence even less appropriate.

State transitions give an exponential model of phone duration, though in practice their contribution to the calculation of the joint probability of state and observation sequences is overpowered by that of the output distribution. In fact, Merhav & Ephraim (1991) show that an HMM with a continuous output distribution, such as typically used for ASR, converges asymptotically to an independent process as the dimension of the feature vector increases. Another independence assumption which is normally made by HMMs is spatial. To reduce parameterization and increase efficiency of computation, Gaussian mixture components have diagonal covariance matrices. Dependencies persist between feature dimensions even for front-end parameterizations which are designed to decorrelate the frequency components of the speech signal, such as Mel-frequency cepstral coefficients (MFCCs), which are described in Section 3.1.2. Diagonal covariance matrices

[2]The approximation of finding the single most likely state sequence rather than summing over all possible sequences is described in more detail in Section 6.1.1 on page 187.

cannot model such correlations.

Dealing with phonetic and phonological variation in HMM systems involves the use of large numbers of context-dependent models. It is not unusual to employ biphone, triphone, quadriphone, and even quinphone or higher models. Taking triphones as an example, where immediate left and right contexts are accounted for, different models would exist for the [`ae`] in [`s ae t`] and in [`h ae t`] [3]. Context-dependent models dramatically improve recognition performance, though at the cost of data-sparsity problems for all but the largest corpora. Considering that English is typically described with approximately 46 phones, building a system of triphone models would require estimating the parameters of $46^3 = 97336$ models, less the combinations which do not occur due to phonotactic constraints.

Training then becomes an exercise in reduction of the number of free parameters. The number of triphones can be restricted by only using them for word internal positions (word internal) rather than in every instance (cross-word), and data-driven parameter tying schemes are usually employed to group together similar states which then share training data and parameters. A criticism of this approach to contextual modelling for HMMs has been that no account is taken of the source of the variation, though there have been some linguistically motivated approaches. These are described in Section 2.2.2 on page 28.

The frame-level Gaussian mixtures used for acoustic modelling in standard HMMs give an unstructured representation of the acoustic parameters. The model is free to switch mixture components at every frame, providing a means of modelling acoustic trajectories without knowledge of the underlying process (Iyer, Gish, Siu, Zavaliagkos & Matsoukas 1998). Furthermore, factors which may confuse phonetic identity such as speaker characteristics or environmental noise must all be accounted for within the one distribution, regardless of the time-scale over which they occur.

Despite these criticisms, the HMM is the model which made large vocabulary speech recognition possible. Indeed, it is the simplifying assumptions which make HMM training

[3] Throughout this thesis, the phonetic alphabet used will be the ARPABET-like systems used to label the TIMIT and MOCHA corpora. Appendix A.1 gives the mappings of these two phone sets to IPA symbols.

and decoding so efficient.

1.3.2 Hybrid ANN/HMM

In standard HMMs, Gaussian mixture models provide the means of computing the probability of an observation $\mathbf{y}_t$ given a state $q_j \in \mathcal{Q}$, $p(\mathbf{y}_t|\, q_j)$. This quantity is turned around in a hybrid ANN/HMM approach in which an artificial neural network is trained to give a direct estimate of the posterior probability of state q_j given an observation, $p(q_j|\,\mathbf{y}_t)$. Using Bayes' rule, the two can be related, giving

$$p(q_j|\,\mathbf{y}_t) = \frac{p(\mathbf{y}_t|\, q_j)\, p(q_j)}{p(\mathbf{y}_t)} \tag{1.2}$$

Models are compared with a *scaled likelihood* (Morgan & Bourlard 1995, Robinson, Cook, Ellis, Fosler-Lussier, Renals & Williams 2002) where $p(\mathbf{y}_t)$ is ignored since it is independent of the state:

$$p(\mathbf{y}_t \,|\, q_j) \propto \frac{p(q_j \,|\, \mathbf{y}_t)}{p(q_j)} \tag{1.3}$$

In practice, the ANN takes a number of adjacent acoustic frames as input, so that $\mathbf{y}_t$ is replaced with $\mathbf{y}_{t-\tau}^{t+\tau}$. Frames surrounding the time of interest are included in any likelihood calculation, thereby loosening the standard HMM assumption of framewise independence. This brings an implicit model of context dependency into the acoustic modelling as observations from neighbouring phones will appear in the input near model boundaries. The HMMs for which the ANN provides emission probabilities typically have simple topologies, and are used to impose minimum durations and give phone transition probabilities. For example, the HMMs used in Robinson et al. (2002) are single state monophone models.

Artificial neural networks provide general non-linear mappings which make minimal assumptions regarding functional form, and have the capacity to model any dependencies present in the data. However, ANNs tend to be treated as 'black box' classifiers as meaningful analysis of their inner workings is difficult. Both spatial and inter-frame correlations occurring within the input window can be accounted for, and recurrent neural networks have also been used to introduce explicit models of time dependency. An ANN cannot really be seen as providing a model of speech production, though the ability to model a multitude of dependencies and use of discriminative training makes them powerful classifiers.

Training an ANN once needed specialist hardware, but now with the availability of cheap computing, the computational demands of parameter estimation can be met on standard systems. Once trained, mapping from input to target domain with a neural network is extremely rapid. This ease of evaluation, combined with the need for relatively few models makes the hybrid ANN/HMM an extremely efficient manner in which to implement ASR, both in terms of memory and CPU usage. Robinson (1994) reports the current lowest phone error rate for TIMIT phone recognition using a hybrid ANN/HMM system.

1.3.3 Segment models

In the context of speech recognition, a segment refers to a variable-length section of acoustic data, which usually corresponds to some linguistic unit. Rather than modelling speech at the frame level, such as in HMMs, or with a fixed context window as in hybrid ANN/HMM systems, segment models aim to characterise entire, usually linguistically meaningful, sections of speech. Such models have the possibility of capturing dependencies which occur over a variety of time-scales, whether intra-frame, between consecutive frames or across entire segments. Furthermore, there is no requirement that all segment models use identical acoustic analysis. It may be that feature extraction could usefully be varied to suit the nature of each segment type.

Distinct sources of variation One advantage of segment modelling is the potential to separate out two types of variation which an acoustic model must deal with. Supposing that model $m \in \mathcal{M}$ characterises a segment with some target distribution $t_m \in \mathcal{T}$, external factors such as speaker characteristics and context will affect the realisation of t_m. This is known as *extra*-segmental variation, and applies to the entire duration of the segment. Once this has been accounted for, there will be an *intra*-segmental variation of the observations around the target t_m. Models of frames or input windows are forced to model these two kinds of variation together, though current systems use speaker adaptation to provide a *post-hoc* solution to dealing with inter-speaker variability

(Young, Evermann, Kershaw, Moore, Odell, Ollason, Povey, Valtchev & Woodland 2002).

Segment modelling can, and has, taken a multitude of different forms, and a review of those which have been applied to ASR is given in Section 2.3 on page 33. The work reported in this thesis revolves around the application of a segment model, the linear dynamic model (LDM), which will be introduced in Section 2.1.2 on page 20 and described fully in Chapter 4.

One of the acoustic modelling ideals listed above was to reflect the nature of the production mechanism. This could be achieved by building a model in which the internal states are derived from the properties of the articulators. The following section considers the ways in which knowledge of the underlying articulation might aid speech recognition.

1.4 Articulation as an information source for ASR

Human speech recognition is a highly developed process. Most people manage a high level of accuracy, despite constantly having to switch between (potentially unfamiliar) speakers and a variety of noise conditions. Those who subscribe to the motor theory of speech perception (Liberman & Mattingly 1985) would argue that 'we can, because we know how'. Using articulatory information in ASR offers the chance to explicitly model some of the effects which arise from simple variation in the production mechanism, yet cause significant changes to the acoustic parameters.

Explicit modelling of phonological variation Speech is produced as a series of articulatory gestures, organised in such a way that movements needed for adjacent vowels and consonants are often made at the same time. The overlapping and asynchronous nature of articulatory gestures allows speech to be produced smoothly and rapidly. Crucially for acoustic modelling of speech, the relatively slowly varying nature of the articulators means that the acoustic information for a given phone will often spread beyond the boundaries of that phone. This appears as co-articulation, which can be defined as *the systematic variation of a speech sound according its context.* An example of co-articulation affecting the acoustic qualities of a phone is the [`i`] in [`i n`], which shows anticipatory nasalisation.

This occurs as during the [i], the velum lowers to be ready for the following [n]. The effect of co-articulation varies considerably dependent on the phone type and the context in which it occurs, though a slight and predictable alteration of an articulatory gesture can have a significant effect on the acoustic realisation of a given phone.

Co-articulation is an inherently articulatory phenomenon. Its effects can most naturally be modelled in the articulatory domain, rather than with a spectral representation where its effects contain an added layer of complexity. An explicit model of co-articulation, an occurrence which provides a significant source of systematic variation, should be beneficial to acoustic modelling for ASR.

It is natural that speech researchers have generally concentrated on modelling an acoustic representation of speech. The data is easy to collect, and after all nothing more invasive than a microphone can be used as the input to a speech recognizer. However, articulatory information can prove useful in complementing a standard acoustic parameterization. Section 2.2 on page 23 describes a variety of approaches which appear in the literature that follow this path. In this thesis, articulatory parameters are used as input features. It is expected that they will prove beneficial in some respects, such as for distinguishing stops, but also lack certain crucial cues. Taking Electromagnetic Articulograph (EMA) data (described on page 48) as an example, making a voiced/voiceless decision or spotting silences is difficult from what is essentially a silent movie of speech. However, the intention will be to use measured articulatory parameters to improve acoustic-only results, and further to see if these results can be replicated using articulation which has been automatically recovered from the acoustics.

The articulatory parameters are considered a design tool which give a real-world basis on which to work and a source of inspiration for the properties which an acoustic model for ASR should possess. A model of measured human articulation will not ultimately be useful in a speech recognition system, though understanding the characteristics of such data will be. Separate recovery of articulation and subsequent modelling of these parameters, if shown to be advantageous, should be combined. Training both parts of this process together would render the acoustic-articulatory mapping into an internal state of the acoustic model.

1.5 Motivation for this thesis

This thesis is motivated by the belief that a model which reflects the characteristics of speech production will ultimately lead to improvements in automatic speech recognition performance. The articulators move slowly and continuously along highly constrained trajectories, each one capable of a limited set of gestures which are organised in an overlapping, asynchronous fashion. Feature extraction on the resulting acoustic signal produces a piecewise smooth, spatially correlated set of parameters in which neighbouring feature vectors are highly correlated, and dependencies can spread over many frames. These properties should be reflected in an acoustic model.

This work investigates speech modelling using a form of linear state-space model, and the related implementational issues. The intention is to show that a hidden dynamic representation can be used to enhance speech recognition. Furthermore, the possibility of acoustic modelling in which an underlying representation is continuous both within and between phones would allow modelling of longer range dependencies, and loosen the standard assumption of inter-segmental independence.

The remainder of the thesis is organised as follows: Chapter 2 provides a review of the literature relevant to this work, divided into sections which deal with linear Gaussian models, articulatory-inspired approaches to speech recognition, and speech recognition using segment models. Chapter 3 then describes the data and some of the basic techniques required for the experimental work in this thesis. The final section of this chapter describes some preliminary analysis of the relevant acoustic and articulatory data which examines the suitability or otherwise of linear predictors of the dependencies between and within phone segments. Chapter 4 gives an in-depth look at linear dynamic models, their properties, parameter estimation, model evaluation and the assumptions made in applying such models to speech data. Chapter 5 presents a series of classification experiments which are designed to assess the contribution which is made by the inclusion of a dynamic state against static model baselines for a variety of data-sets. This chapter then goes on to consider some variations on the implementation of linear dynamic models. Chapter 6 describes the time-asynchronous A^* search strategy which has been used to implement decoding with LDMs, and presents the results of full recognition experiments. Chapter 7 summarises the findings of this thesis and suggests directions for future work.

1.6 Publications

A number of publications have resulted from work during the period of study for this thesis. These are: Frankel, Richmond, King & Taylor (2000), King, Taylor, Frankel & Richmond (2000), Frankel & King (2001*a*) and Frankel & King (2001*b*).

Chapter 2

Literature Review

A survey of the literature relevant to this thesis falls neatly into three parts: linear Gaussian models, speech recognition which incorporates articulatory information, and segmental approaches to acoustic modelling. Linear Gaussian models will be dealt with first, as they will prove useful in describing some of the segment models of Section 2.3.

2.1 Linear Gaussian models and their relatives

Linear Gaussian models are a class of models which have found many applications in the last few decades, in domains such as control, machine learning and financial analysis. There are two excellent papers offering reviews of such models, Roweis & Ghahramani (1999) and Rosti & Gales (2001), which are drawn from for this summary.

2.1.1 State-space models

As a starting point, it is useful to introduce the idea of a state-space model, in which data is described as a realisation of some unseen, usually lower dimensional, process. The following two equations describe a general state-space model:

$$\mathbf{y}_t = h(\mathbf{x}_t, \boldsymbol{\epsilon}_t) \tag{2.1}$$

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \ldots, \mathbf{x}_1, \boldsymbol{\eta}_t) \tag{2.2}$$

A p-dimensional observation $\mathbf{y}_t$ is linked to a q-dimensional state vector $\mathbf{x}_t$ by means of Equation 2.1, and the state's evolution is governed by Equation 2.2.

The observation noise $\boldsymbol{\epsilon}_t$ characterises the variation due to a range of external sources, for example measurement error or noise. Furthermore it offers a degree of smoothing which is useful when there is a mismatch between training and testing data. Uncertainty in the modelling of the state process is described by the state noise $\boldsymbol{\eta}_t$. An important feature of these models is that the observation at time t is conditional only on the state at that time. However, the state can take a variety of forms, such as static distributions, long-span auto regressive processes or sets of discrete modes. Figure 2.1 represents such a model, where motions in the state space give rise to the observed data.

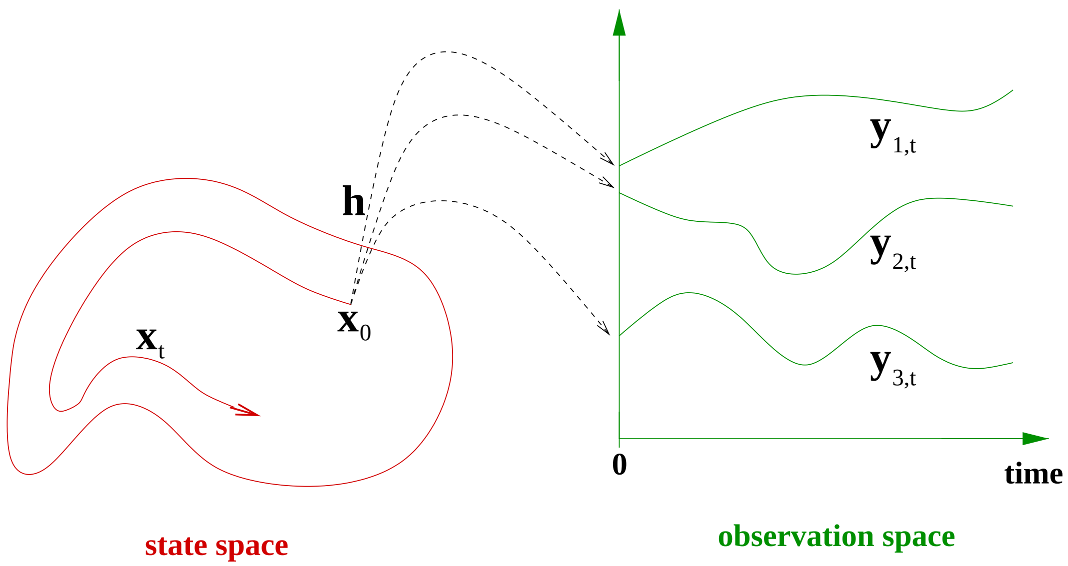

Figure 2.1: In state-space models, the observations are seen as realisations of some unseen, usually lower-dimensional, process. This provides a means of distinguishing the underlying system from the observations which represent it. The state and observation spaces are linked by the transformation $\mathbf{h}$.

State-space models are useful in many real-life situations where systems contain a different number of degrees of freedom, usually fewer, than the data used to represent them. In these cases, a distinction can be made between the production mechanism at work and the parameterization chosen to represent it. The hidden state variable can have just as many degrees of freedom as are required to model any underlying processes, and then a state-observation mapping shows how these are realised in observation space. This offers a means of making a compact representation of the data. In fact, dimensionality reduction is a common application of this class of models (Carreira-Perpiñán 1997).

There are two problems which must usually be solved for practical application of any given state-space model. Firstly, it should be possible to infer information about the internal states of the model for a given set of parameters and sequence of observations. Secondly, the parameters which identify the model must be estimable given suitable training data.

Inference

For a fixed set of model parameters Θ, and an N-frame sequence of observations $\mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, there are a number of quantities which may be calculated. The total likelihood of a set of observations can be computed as the integral of the joint probability of state and observations $p(\mathbf{y}_1^N, \mathbf{x}_1^N|\Theta)$ over all possible state sequences. With $\mathbf{x}_1^N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ representing one such candidate state sequence, the integral to be evaluated is

$$p(\mathbf{y}_1^N|\Theta) = \int_{\text{all } \mathbf{x}_1^N} p(\mathbf{y}_1^N, \mathbf{x}_1^N|\Theta)\, d\mathbf{x}_1^N \tag{2.3}$$

An application of Bayes' rule then provides the conditional probability of a given state sequence underlying the observations:

$$p(\mathbf{x}_1^N|\mathbf{y}_1^N, \Theta) = \frac{p(\mathbf{y}_1^N, \mathbf{x}_1^N|\Theta)}{p(\mathbf{y}_1^N|\Theta)} \tag{2.4}$$

Filtering and smoothing are operations which lie at the heart of inference calculations for all but the simplest of state-space models. Filtering produces an estimate of the state distribution at time t given all the observations up to and including that time, $p(\mathbf{x}_t|\mathbf{y}_1^t, \Theta)$, and smoothing gives a corresponding estimate of the state conditioned on the entire N-length observation sequence, $p(\mathbf{x}_t|\mathbf{y}_1^N, \Theta)$. The Kalman filter and Rauch-Tung-Striebel (RTS) smoother provide the optimal solutions for linear Gaussian models and are detailed in Section 4.2.1. Equivalent techniques for filtering/smoothing of non-linear or non-Gaussian models exist, though optimal solutions are not common. See Haykin (2001) for details of the extended Kalman filter, particle filtering and so on.

Estimation

A hidden state adds a degree of complication to parameter estimation. Before describing the expectation maximisation (EM) algorithm, which offers a general framework for dealing with cases which have missing data (such as a hidden state sequence), standard maximum likelihood is outlined.

Maximum likelihood If the state were observable, it would be possible to find standard maximum likelihood (ML) estimates of the parameters. The state and observation data are regarded as fixed, and the parameter set Θ is treated as a random variable. Writing $\mathcal{Y} = \mathbf{y}_1^N$ and $\mathcal{X} = \mathbf{x}_1^N$, the likelihood function for the model is

$$L(\Theta|\mathcal{Y}, \mathcal{X}) = p(\mathcal{Y}, \mathcal{X}|\Theta) \tag{2.5}$$

The ML solution corresponds to finding a value of Θ which maximises 2.5. For many models, this maxima can found analytically, often by maximising $\log L(\Theta|\mathcal{Y}, \mathcal{X}) = l(\Theta|\mathcal{Y})$ rather than $L(\Theta|\mathcal{Y}, \mathcal{X})$ directly. However, alternatives to standard ML techniques must be employed when the state is unobserved.

Expectation maximisation algorithm One such approach is the EM algorithm (Dempster, Laird & Rubin 1977) which integrates the incomplete-data log-likelihood $L(\Theta|\mathcal{Y})$ over the missing data, which in this case is the state. Taking logarithms, the likelihood function is

$$\begin{aligned} \log L(\Theta|\mathcal{Y}) = l(\Theta|\mathcal{Y}) = \log p(\mathcal{Y}|\Theta) &= \log \int_{\mathcal{X}} p(\mathcal{Y}, \mathcal{X}|\Theta)\, d\mathcal{X} \\ &= \log \int_{\mathcal{X}} p_0(\mathcal{X}) \frac{p(\mathcal{Y}, \mathcal{X}|\Theta)}{p_0(\mathcal{X})}\, d\mathcal{X} \end{aligned} \tag{2.6}$$

where $p_0(\mathcal{X})$ is an arbitrary distribution over the state variables. Since $\int_{\mathcal{X}} p_0(\mathcal{X}) = 1$, and the logarithm function is concave ($d^2x/dx^2 \log x = -x^{-2} < 0 \;\; \forall x \in \Re$), Jensen's inequality can be applied to give a lower bound on 2.6:

$$l(\Theta|\mathcal{Y}) \geq \int_{\mathcal{X}} p_0(\mathcal{X}) \log \frac{p(\mathcal{Y}, \mathcal{X}|\Theta)}{p_0(\mathcal{X})}\, d\mathcal{X}. \tag{2.7}$$

The rearrangement of 2.6 to the inequality 2.7 serves two purposes. Firstly, the original likelihood function contained the log of an integral. In 2.7, the log has been moved within the integral which greatly simplifies analytical optimisation. Secondly, choosing

$p_0(\mathcal{X}) = p(\mathcal{X}|\mathcal{Y}, \Theta^{(i)})$, the posterior of the state given the observations $\mathcal{Y}$ and some previous model parameters, $\Theta^{(i)}$, gives equality in 2.7. In cases where $p(\mathcal{X}|\mathcal{Y}, \Theta^{(i)})$ is unavailable or intractable, alternate distributions can be used for $p_0(\mathcal{X})$, and maximising the expression in the right hand side of 2.7 will still increase, or at worst not affect, the incomplete-data likelihood function $\log L(\Theta|\mathcal{Y})$.

For many classes of model, including all those dealt with in this section, the state posterior distribution is available. Since the denominator on the right hand side of 2.7 is not dependent on Θ, and setting $p_0(\mathcal{X}) = p(\mathcal{X}|\mathcal{Y}, \Theta^{(i)})$, maximising $\log L(\Theta|\mathcal{Y})$ is equivalent to maximising what is known as the auxiliary function:

$$Q(\Theta, \Theta^{(i)}) = \int_{\mathcal{X}} p(\mathcal{X}|\mathcal{Y}, \Theta^{(i)}) \log p(\mathcal{Y}, \mathcal{X}\,|\Theta)\, d\mathcal{X}. \tag{2.8}$$

$$= E\left[\log p(\mathcal{Y}, \mathcal{X}\mid \Theta) | \,\mathcal{Y}, \Theta^{(i)}\right] \tag{2.9}$$

$$= E_{\Theta^{(i)}}\left[\, l(\Theta\,|\mathcal{Y}, \mathcal{X}\,)|\,\mathcal{Y}\,\right] \tag{2.10}$$

This maximisation is carried out in two stages. Given some starting parameter estimates, the goal of the E-step is to:

> *find $Q(\Theta, \Theta^{(i)})$, the expectation of the complete-data log-likelihood with respect to the unknown data $\mathcal{X}$, given the observed data $\mathcal{Y}$, using the most recent parameter estimates $\Theta^{(i)}$, i.e. evaluate*
>
> $$Q(\Theta, \Theta^{(i)}) = E_{\Theta^{(i)}}\left[\, l(\Theta|\mathcal{Y}, \mathcal{X}\,)|\mathcal{Y}\right] \tag{2.11}$$

Then in the M-step:

> *expectations computed in the E-step are used to maximise model parameters giving*
>
> $$\Theta^{(i+1)} = \underset{\Theta}{\mathrm{argmax}} Q(\Theta, \Theta^{(i)})$$

Less formally, in the E-step, estimates are made of the state values using the most recently estimated parameter set and the observation sequence. These are then used in place of actual values in the standard ML solutions in the M-step. The EM algorithm combines these operations and steps iteratively toward the ML solution. The parameters must be initialised to some sensible starting value as training is dependent on initial conditions. This is because whilst the EM algorithm guarantees an increase in likelihood at each iteration, it does not guarantee to find the global maximum. Derivation of the EM algorithm for linear dynamic models is given in Section 4.2.2.

2.1.2 Linear Gaussian models

In this class of models, the transformations H and F are restricted to being linear mappings, and the error terms $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are additive Gaussian noise. Furthermore, if the state is dynamic, the initial state must also be specified and is also Gaussian. This set of constraints ensures that the distribution over both state and output processes are always Gaussian. Modifying 2.1 and 2.2 accordingly, a linear Gaussian model is described by

$$\begin{aligned} \mathbf{y}_t &= H_t \mathbf{x}_t + \boldsymbol{\epsilon}_t & \boldsymbol{\epsilon}_t &\sim N(\mathbf{v}, C) \\ \mathbf{x}_t &= \sum_{j=1}^{k} F_t^{(j)} \mathbf{x}_{t-j} + \boldsymbol{\eta}_t & \boldsymbol{\eta}_t &\sim N(\mathbf{w}, D) \end{aligned}$$

and an initial state distribution $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$. For our purposes, H and F are time-invariant, and the state process is at most 1^{st} order, so that $k = 1$. A linear Gaussian model can be categorised as static or dynamic, dependent on the properties of its state process.

Static models

In a static model, the temporal ordering of the data is ignored. Setting $F = \mathbf{0}$ removes the dynamic portion of the model and the state is simply modelled as a Gaussian. Equations 2.12 and 2.13 describe such a model.

$$\mathbf{y}_t = H_t \mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{2.12}$$

$$\mathbf{x}_t = \boldsymbol{\eta}_t \qquad \boldsymbol{\eta}_t \sim N(\mathbf{w}, D). \tag{2.13}$$

Given that the output distribution is simply represented by a single static Gaussian distribution, the advantage of such a model over a standard Gaussian is not immediately apparent. However, the aim is to absorb the correlation structure of the data into H, and constrain the observation noise covariance C to be diagonal or some multiple of the identity I. In this way, high dimensional data can be described with a minimum number of parameters.

The degeneracy present (see LDMs on page 20 below) in the model means that the state noise distribution can be restricted to $\boldsymbol{\eta}_t \sim N(\mathbf{0}, I)$ with no loss of generality. The observation noise mean $\mathbf{v}$ takes the mean of the data, and the form of the covariance C will be discussed in the sections below. Noting that D has been set to be an identity matrix, the output distribution can be found using analytical integration:

$$\mathbf{y}_t \sim N(\mathbf{v}, HH^T + C) \tag{2.14}$$

and inference using linear matrix projection:

$$\mathbf{x}_t|\mathbf{y}_t \sim N(\boldsymbol{\beta}\mathbf{y}_t, I - \boldsymbol{\beta}H), \tag{2.15}$$

where $\boldsymbol{\beta} = H^T(HH^T + C)$. Likelihood computation under such models simply involves evaluation of the Gaussian in 2.14, and EM parameter estimation is straightforward and detailed in both of the review papers mentioned above.

Sensible principal components analysis Standard principal components analysis (PCA) sets C to zero, and so does not in fact describe a density function over the observation space. This means that it is not strictly a linear Gaussian model, though is a widely used dimensionality reduction tool in which the columns of H span the principal subspace of the data. However, sensible principal components analysis (SPCA) (Roweis 1999) puts a spherical Gaussian covariance onto the observation noise $\boldsymbol{\epsilon}_t$, whilst still characterising the data in terms of a subspace where the columns of H are known as the principal components. These are chosen to maximise the variance retained in the lower-dimensioned projection of the observations.

Factor analyser In a factor analysis (FA) model, the observation noise C is restricted to being diagonal. This gives a Gaussian output distribution over the observations though uses only $2p + pq$ parameters compared to the $p + p(p+1)/2$ parameters in a standard p-dimensional Gaussian. If C were not constrained in this way, parameter estimation could simply place zeros in H and set $\mathbf{v}$ and C to be the sample mean and covariance. This would amount to a valid maximum likelihood solution, though not a very informative one. The power of a factor analysis model lies in the possibility of providing a compact representation of high dimensioned data. Indeed, if there is enough training data to give full rank estimates of the sample covariance, there is nothing to be gained using factor analysis over a straightforward Gaussian distribution.

Dynamic models

Linear dynamic model The linear dynamic model (LDM) is described in some detail in Chapter 4, though for continuity a quick introduction is given here. The static models above aim to characterise the spatial correlations of the data, and the LDM builds on this by also modelling the temporal characteristics of the data. An LDM is described by the following pair of equations:

$$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{2.16}$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t \qquad \boldsymbol{\eta}_t \sim N(\mathbf{w}, D) \tag{2.17}$$

and a distribution over the initial state, $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$. Equation 2.17 describes how a Gaussian density flows through a continuous state-space, transformed according to the rotations and stretches of F and the addition of noise. The state distribution is directly related to the observations by the linear mapping given in (2.16).

Degeneracy The LDM is an over-specified model. For a model with observation dimension p, state dimension q, full covariance matrices and non-zero noise terms, there are

$$p + p(p+1)/2 + pq + q^2 + q + q(q+1)/2 + q + q(q+1)/2 \tag{2.18}$$

parameters. These correspond to the observation noise mean and covariance, observation matrix, state evolution matrix, state noise mean and covariance, and finally initial state mean and covariance. However, as demonstrated in Roweis & Ghahramani (1999), the structure of the state noise covariance, D, can be incorporated into F and H. D can therefore be restricted to being an identity matrix with no loss of generality, and so the

LDM in fact has

$$p + p(p+1)/2 + pq + q^2 + 2q + q(q+1)/2 \tag{2.19}$$

free parameters. A unique solution is not guaranteed, as re-ordering the corresponding columns of H and F whilst also swapping the dimensions of the state will leave the output distribution unaffected.

2.1.3 Non-linear and/or non-Gaussian extensions

There are a number of commonly used state-space models which will not feature in this thesis, though a quick summary will serve to show the linear models in context. Either or both state and observation processes can be varied, and below are some of the possibilities.

Variations on the state process

The continuous state can be replaced with a discrete process, so that

$$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{2.20}$$

$$\mathbf{x}_t = WTA[F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t] \qquad \boldsymbol{\eta}_t \sim N(\mathbf{w}, D) \tag{2.21}$$

where the 'winner-take-all' non-linearity $WTA[\mathbf{x}]$ produces a new vector with all elements set to zero apart from one which is set to unity. This state process dictates which of the columns of H is used to generate the mean of the output distribution. This is a particular form of hidden Markov model where the q mixture components share a common covariance. Setting $F = 0$ in 2.21 gives a mixture of q Gaussians model with a pooled covariance, and further setting $C = 0$ in 2.20 produces a vector quantization (VQ) model. Independent component analysis (ICA) for blind source separation is also discussed in Roweis & Ghahramani (1999), where the WTA non-linearity is replaced with a general non-linearity, $g(\mathbf{x})$.

Varying the observation process

The straightforward linear mapping used in all the models so far can be replaced with what Rosti & Gales (2001) term a linear discriminant analysis (LDA) observation process. It is so named since with a static state, the model becomes standard LDA, which aims to give maximal linear separation of the data by class, by minimising within-class variance at the same time as maximising between-class variance. See Section 3.2.1 on page 62 for more details of LDA.

Mixture distributions

In addition to the pooled covariance Gaussian mixture model described above, there are other means of modifying linear Gaussian models to give multimodal distributions over the observations.

Error terms can be replaced with mixture distributions and process parameters chosen from a number of candidates according to an indicator function. Extending the static models in this way provides an efficient method of representing multimodal, non-Gaussian, spatially correlated observations. Parameter estimation and inference for such models can be found in Rosti & Gales (2001). However, equivalent extensions of dynamic-state models are subject to problems of computational intractability: state evolution would require conditioning on a mixture distribution, causing exponential growth in the number of components describing the state. Filtering and smoothing are therefore intractable for such models and approximate methods must be employed. Section 7.3.2 on page 235 outlines some of these, and further information can be found in Ghahramani & Hinton (1996*b*), Murphy (1998), and Rosti & Gales (2001).

2.2 Articulatory-inspired acoustic modelling for ASR

Efforts to incorporate articulatory information into acoustic modelling for speech recognition have been made by a number of researchers in a variety of ways. These include using articulatory parameters directly as features for recognition, recognition by synthesis from articulatory parameters, and articulatory representations to constrain or dictate model structure. Articulatory information used in ASR is parameterized in one of the following two ways:

Discrete In most cases, the articulatory representation consists of an inventory of discrete multi-levelled features which are chosen based on linguistic theory. These are known as *pseudo-articulatory* features and may include categories such as *lip rounding* {+/-} or *tongue body* {back/mid/front}. These are commonly produced using rule-based systems which map from existing labels to a corresponding articulatory configuration or sequence of configurations (Eide, Rohlicek, Gish & Mitter 1993, Kirchhoff 1998, King & Taylor 2000, Chang, Greenberg & Wester 2001), though can also be derived by quantizing measured articulatory data (Stephenson, Bourlard, Bengio & Morris 2000).

Continuous In a few systems, such as Zlokarnik (1995*a*) and Wrench (2001), the articulatory input takes the form of a number of continuous streams of data which together give a smoothly-varying description of the motion of the articulators. These may consist of measured human articulation, or parameters which have been automatically recovered from an acoustic input.

2.2.1 Articulatory parameters as features

The systems described below attempt to improve acoustic-only recognition accuracies through the direct inclusion of articulatory features. This requires a corpus either with parallel acoustic-articulatory data such as MOCHA (Wrench 2001), which will be detailed in Section 3.2.1, or a system for automatically generating articulatory parameters to accompany existing data.

In either case, a practical recognizer can only rely on the acoustic signal for input, and some means of dealing with the absence of articulatory features at recognition time is required. One approach is to treat the articulation as missing data during recognition, and another is to infer articulatory parameters from the acoustic data which are then used in place of the real ones. Recovering articulation from the acoustics is known in speech processing as *articulatory inversion*, and is described in Section 3.1.1. Data generated in this way will be used in experiments later in the thesis, and referred to as 'recovered articulation'.

Real articulatory features

The systems outlined in this section all use features derived from measured human articulation. In all cases, these features are found to aid recognition. However, recovered articulatory features have yet to prove as useful.

HMM systems Zlokarnik (1995*a*, 1995*b*) conducted recognition experiments in which real or automatically recovered articulatory parameters were appended to acoustic features in an HMM recognizer. An electromagnetic articulograph (EMA) as described on page 48 was used to record articulatory traces to accompany the acoustic data for 165 vowel-consonant-vowel (VCV) sequences for a male and a female speaker. These were constructed from 4 vowels and 14 consonants and embedded in a German carrier sentence. The automatically recovered articulatory parameters were produced using a multi-layer perceptron (MLP) trained to map between acoustic and articulatory parameters. The HMMs were implemented with 2 states per phone – one left-context dependent and one context-independent. With the VCV sequences treated as isolated words, adding real articulation to an MFCC representation of the acoustics increased the recognition rates from 89.4% to 96.3% and 82.2% to 93.3% for the male and female speakers respectively. These represent relative error reductions of more than 60%. Replacing real with recovered articulatory parameters also improved recognition performance, giving relative error reductions of 18% and 25% on data from the male and female speakers respectively.

The performance increases were not as dramatic when automatically estimated articulatory parameters were used in place of the real ones. However, the MLP was able to supplement the information in the acoustic parameters. This may be attributable to the supervised nature of training an MLP. If so, the mapping learned between input and target domains allowed the MLP to recover some of the cues that aid word discrimination which are present in the articulatory parameters, but less apparent in the acoustics. Alternatively, the increase in recognition accuracy might be due to contextual information contained in the recovered articulatory parameters, as the MLP used a input layer spanning 51 frames, or 0.51 seconds.

Similar experiments were conducted on a larger scale by the collector of the MOCHA (Wrench 2001) database. The experiments were speaker-dependent, and used 460 TIMIT type sentences recorded by a southern English female speaker. It was found that augmenting an acoustic feature set to include real articulatory parameters gave a 9% relative increase in phone accuracy for continuous recognition with a triphone HMM system (Wrench & Hardcastle 2000, Wrench 2001). The baseline acoustic recognition accuracy was 65% using 14 MFCCs along with their δ and $\delta\delta$ coefficients. An articulatory feature set was generated by stacking the EMA, laryngograph and electropalatograph (EPG) data with their corresponding δ and $\delta\delta$ coefficients and performing linear discriminant analysis (LDA) dimensionality reduction. On the same recognition task as above, the accuracy was 63%, similar yet lower than acoustic-only performance. Combining all acoustic and articulatory data and again using LDA to produce a 45-dimensional feature vector, recognition accuracy for the same task was 71%, higher than for either acoustic or articulatory features used on their own. However, when real articulation was replaced with articulatory parameters automatically generated from the acoustics using a multi-layer perceptron (MLP), there was no improvement over the baseline acoustic result (Wrench & Richmond 2000).

Dynamic Bayesian network system Dynamic Bayesian networks (DBN) are extensions of Bayesian networks for modelling dynamic processes. These models have been applied to standard acoustic speech recognition (Zweig & Russell 1998), though are ideal for incorporating articulatory information as handling missing data is straightforward. Articulation can be used in all or some of the training set, and can be hidden at recognition time.

With $q_t \in \mathcal{Q}$ representing the state at time t, standard HMMs model the probability of each observation $\mathbf{y}_t$ given the current state q_t, along with the probability of transitioning from one state to the next. These are written as:

$$p(\mathbf{y}_t|q_t) \tag{2.22}$$

$$\text{and} \qquad p(q_t|q_{t-1}) \tag{2.23}$$

DBNs provide a framework in which it is straightforward to include other causal relationships. Letting a_t represent an articulator position at time t, Stephenson et al. (2000) conducted recognition experiments in which the state transition 2.23 is modelled along with

$$p(\mathbf{y}_t|a_t, q_t) \tag{2.24}$$

$$\text{and} \qquad p(a_t|a_{t-1}, q_t) \tag{2.25}$$

Under this model, the observations are conditioned on both state and articulator position, shown by 2.24. Furthermore, in 2.25, the new articulatory configurations are conditioned not only the previous one, but also on the current state, providing an element of contextual modelling.

The Wisconsin x-ray microbeam database (Westbury 1994) provided parallel acoustic-articulatory data for an isolated word recognition task. The acoustic features were 12 MFCCs and energy along with their δ coefficients, and the articulatory features consisted of x and y coordinates for 8 articulator positions (upper lip, lower lip, four tongue positions, lower front tooth, lower back tooth). DBNs take discrete observations, and so codebooks were generated for the acoustic and articulatory data-sets using K-means.

The acoustic-only word error rate of 8.6% was reduced to 7.6% when the articulatory data was used during recognition. With the articulation hidden, the system gave a recognition word error rate of 7.8%, which is a 9% relative error decrease over the acoustic baseline.

Pseudo-articulatory features

Using pseudo-articulatory features for acoustic modelling requires a mapping between continuous acoustic and discrete articulatory feature domains. Both HMMs (Eide et al. 1993) and neural networks (King & Taylor 2000, Chang et al. 2001) have been applied to this task.

Kirchhoff The fullest investigation into recognition using articulatory features has been reported by Kirchoff (1998, 2002). One of the main motivations for this work was to build a system which would be robust to noise. Using a phonetic-level transcript, the telephone speech corpus OGI numbers95 (Cole, Noel, Lander & Durham 1995) was marked up with the pseudo-articulatory features shown in Table 2.1 according to a set of canonical phone-

Feature group	Feature values
voicing	+voiced, −voiced, silence
manner	vowel, lateral, nasal, fricative, approximant, silence
place	dental, coronal, labial, retroflex, velar, glottal, high, mid, low, silence
front-back	front, back, nil, silence
rounding	+round, −round, nil, silence

Table 2.1: Pseudo-articulatory features and levels used to mark up the Numbers95 corpus for Kirchhoff's articulatory feature-based recognition system.

feature conversion rules. The conversion mappings can be found in Kirchhoff (1998). A separate MLP was trained to predict the values for each feature based on the acoustic input. Using a further MLP, the outputs from these 5 networks were mapped to phone class posteriors which could be used in a standard hybrid HMM/ANN recognition formulation.

On clean speech, the word error rates for the acoustic and articulatory models were comparable, 8.4% and 8.9% respectively, though in the presence of a high degree of additive noise, the articulatory model produced significantly better results. At a noise level of $0dB$[1], the word error rate for the acoustic model was 50.2%, higher than the 43.6% produced under the articulatory system. When the outputs of the acoustic and articulatory recognizers were combined, the error rates were lower than for either of the two individually under a variety of noise levels and on reverberant speech. The framewise errors for the different articulatory feature groups show that classification performance on the voicing, rounding and front-back features does not deteriorate as quickly as for manner and place in the presence of noise. This appears to support the author's claim that a system of 'experts' where each MLP is only responsible for distinguishing between a

[1] The relative intensity (sound energy) of two signals is measured in decibels (dB): $L = 10 \log_{10} |I_1/I_2|$ (Gold & Morgan 1999). Thus 0dB signifies a signal and noise with equal intensity.

small number of classes would be more robust to adverse conditions than one 'monolithic' classifier. By incorporating confidence scores when the outputs of individual classifiers are combined, the system could be tailored to particular operating conditions.

Similar experiments were performed on a larger spontaneous dialogue corpus. Verbmobil (Kohler, Lex, Patzold, Scheffers, Simpson & Thon 1994) contains 31 hours of training data and 41 minutes of test data from a total of 731 speakers. Once again, improvements were shown when acoustic and articulatory features were combined. The word error rate in this case was 27.4%, a 6% relative error reduction on the acoustic baseline of 29.0%.

King King, Stephenson, Isard, Taylor & Strachan (1998) and King & Taylor (2000) also report recognition experiments based on the combination of the output of a number of independent neural network classifiers. The work was primarily aimed at comparing phonological feature sets on which to base the classifiers, though the feature predictions were also combined to give TIMIT phone recognition results. Unlike Kirchhoff who used a neural network to combine the independent feature classifiers, the predicted feature values were used as observations in an HMM system. The resulting recognition accuracy of 63.5% was higher than the result of 63.3% found using standard acoustic HMMs, though the increase was not statistically significant. The need for an asynchronous articulatory model was demonstrated using classifications of a set of binary features derived from Chomsky & Halle (1968). In cases where features changed value at phone boundaries, allowing transitions within two frames of the reference time to be counted as correct, the percentage of frames where all features were correct rose from 52% to 63%. Furthermore, the accuracy with which features were mapped onto the nearest phone rose from 59% to 70%. This demonstrates the limiting nature of forcing hard decisions at phone boundaries onto asynchronous data. In both King and Kirchhoff's systems, the individual feature classifiers were independent. Using a neural network to map features to phone posterior probabilities in the latter gave an implicit model of asynchrony.

2.2.2 Using articulatory parameters to derive HMM topology

Phones are almost always used as the sub-word units for speech recognition. They provide a useful means of describing the structure of language, and the availability of phonetic lexica means that they are a convenient choice for recognizers. However, the realisation of phones varies considerably according to the context in which they occur. Section 1.4 described the increased parameterization which accompanies modelling contextual variation in standard HMM systems. Rather than building vast numbers of models and then reducing parameter numbers by tying states, efforts have been made to use articulatory knowledge to build compact sets of states which still include contextual information where necessary.

HAMM Richardson et al. (2000*a*, 2000*b*) drew on the work by Erler & Freeman (1996) in devising the hidden articulator Markov model (HAMM), which is an HMM where each articulatory configuration is modelled by a separate state. The state transitions aim to reflect human articulation: static constraints disallow configurations which would not occur in American English, and dynamic constraints ensure that only physically possible movements are allowed. Furthermore, asynchronous articulator movement is allowed as each feature can change value independently of the others. In addition to the static constraints which reduced the number of states from $25,600$ to $6,676$, the number of parameters was further reduced by removing states with low occupancy during training.

The recognition task was PHONEBOOK, an isolated word, telephone speech corpus. With a 600 word lexicon, the HAMM gave a significantly higher word error rate than a standard 4-state HMM. These were 7.56% and 5.76% respectively. However, a combination of the models gave a word error rate of 4.56%, a relative reduction of 21% on the HMM system.

Deng Deng and his group (1994*a*, 1994*b*) have also worked on building HMM systems where each state represents an articulatory configuration. Following Chomsky's theory of distinctive features and a system of phonology composed of multi-valued articulatory structures (Browman & Goldstein 1992), they have developed a detailed system for deriving HMM state transition networks based on a set of 'atomic' units. These units represent all combinations of a set of overlapping articulatory features possible under a set of hand-written rules.

Five multi-levelled articulatory features are used: lips, tongue blade, tongue dorsum, velum and larynx. Within each, a '0' level is included and used to indicate when the feature is irrelevant to the specification of a phone. Each phone is mapped to a static articulatory configuration, apart from affricates which are decomposed into stop and fricative portions, and diphthongs which are made up by concatenating appropriate pairs of vowels. Features can spread according to the relative timing of the phone, by 25%, 50%, 75%, or 100%. When the spread is by 100%, a feature can extend as far as the boundary of the next phone in either direction. Then, depending on the configuration of the following phone, the feature might spread further. In this way, long span dependencies can be modelled. When articulatory feature bundles overlap asynchronously, new states are created for the intermediate portions which either describe the transitions between phones or allophonic variation.

On a TIMIT classification task, HMMs constructed from these units achieved an accuracy of 73% compared with context-independent HMMs of phones which gave an accuracy of 62%. The feature-based HMMs also required fewer mixture components, typically 5, than standard phone-based HMMs use. This suggests that a principled approach to state selection will require fewer parameters and therefore less training data, as each state is modelling a more consistent region of the acoustic signal.

This work was developed to include higher level linguistic information by Sun, Jing & Deng (2000). This included utterance, word, morpheme and syllable boundaries, syllable onset, nucleus and coda, along with word stress and sentence accents. This time, results were reported on TIMIT phone recognition, rather than classification. A recognition accuracy of 73.0% was found using the feature-based HMM, which compares favourably to a baseline triphone HMM which gave an accuracy of 70.9%. This represents a 7.2% relative decrease in error.

2.2.3 Recognition by articulatory synthesis

Blackburn (1995, 1996, 1996, 2000) saw the need to model contextual effects in the speech signal in an efficient manner as crucial to the development of speech recognition. His thesis centred around the investigation of an *articulatory speech production model* (SPM) which enabled modelling of co-articulation in the time-domain. Experiments using real articulatory data were carried out on the University of Wisconsin (UW) x-ray microbeam data (Westbury 1994), and further work used the resource management (RM) (Price, Fisher, Bernstein & Pallett 1988) corpus. The system took output from HTK (Young 1993), an HMM recognizer, and performed N-best rescoring by re-synthesising articulatory traces from each time-aligned phone sequence and mapping these into log-spectra using MLPs. Errors between these and the original speech data were calculated and used to re-order the N-best list.

In performing the re-synthesis, the assumption is made that the articulatory gestures which make up each phone can be divided into two parts. In the first, the articulators move away from the configuration corresponding to the previous phone, and in the second, they move toward the configuration needed for the following one. Furthermore, there is some region in which the articulatory configuration gives rise to the sound corresponding to the current phone. This may or may not be static, and is assumed to fall roughly in the middle of the phone.

The SPM models the position of each articulator at the centre of each phone model using a Gaussian distribution, $P \sim N(\mu_P, \sigma_P^2)$. A notion of the articulatory effort required to produce each phone by each of the articulators is introduced by computing the curvatures (2nd-order derivatives) of linearly-interpolated mean articulator positions over a set of time-aligned phone sequences. These are also modelled using Gaussian distributions, $C \sim N(\mu_C, \sigma_C^2)$, and combined with the original articulator-phone positional distribution to arrive at a distribution of position conditioned on curvature for each phone and articulator, $P|C$.

Computing the most likely articulator positions for a sequence of time-aligned phones therefore consists of linearly interpolating the unconditional articulator position means, computing the curvatures for each one, and arriving at new means using the conditional distribution. The end result is a complete set of articulatory traces which incorporate the strength of contextual effects. There was a significant reduction in mean square error between the real and recovered articulation on the UW test set when the co-articulation modelling was included, compared to a baseline where the initial phone-articulator predictions were not adjusted.

The mapping from the re-synthesised articulation to line-spectral acoustic features was handled with a separate MLP for each phone. On the UW corpus, recognition performance was enhanced for all but one of the speakers in the test set using N-best lists with $2 \leq N \leq 5$, however for higher N, the increases on some speakers were offset by decreases for others. The SPM made most contribution to the recognition accuracy for speakers for which there was poor initial performance.

Experiments were also conducted on the RM corpus. Here, N-best rescoring for small N offered modest gains, though performance deteriorated with $N = 100$. However, combining the HTK and SPM output probabilities in the log domain gave relative error reductions of 6.9% and 6.0% for two of the speakers, suggesting some value in the addition of an articulatory re-synthesis stage.

2.3 Segment modelling for speech

Some of the early attempts at speech recognition used what were essentially segment models, though these were rule-based rather than probabilistic models. The last 15 years have seen steady and continued interest in statistical segment modelling from a variety of groups and researchers. However, despite the theoretical advantage of relaxing HMM independence assumptions, segment models have yet to make it into mainstream ASR.

There is a key design choice that is made in any segment model implementation, which is whether the models are of fixed-length or variable-length sequences of observations.

Variable-length The segment model can take observation sequences with a range of durations. Decoding is straightforward as all complete hypotheses account for an equal number of frames.

Fixed-length A pre-chosen fixed-length representation is used for all segments. In some instances this is required by the model which will characterise each segment type. For example, neural networks have fixed-length input windows. Observation sequences are time-normalised before application of the acoustic model. This complicates decoding as candidate hypotheses consisting of different numbers of segments compute their likelihoods over different numbers of frames. A means of normalisation must be included in the decoding strategy.

The methods below use models which aim to capture the temporal correlations within segmental units. In some cases, such as the ANN methods of Section 2.3.5 or the dynamical system model of Section 2.3.6, the models also have the capacity to model spatial correlations across feature dimensions.

2.3.1 Segmental HMMs

By the early 1990s, HMMs were the dominant acoustic model for speech recognition. Recognizing both the efficiency of the HMM framework and the limitations of a frame-based model, Russell (1993) introduced the segmental HMM. Under this model, there is a target t_j associated with each state $q_j \in \mathcal{Q}$ which may be a static Gaussian distribution (Russell 1993), or a linear trajectory (Holmes & Russell 1995). Observations are assumed to be independent given each state q_j, though the output distribution is conditioned on an extra-segmental distribution $p(t_j)$ which remains fixed for the entire state occupancy. Since t_j represents the target associated with state q_j, the joint probability of an observation sequence $\mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and a state q_j can be written as:

$$p(\mathbf{y}_1^N, q_j) = p(t_j)\, p(\mathbf{y}_1^N \,|\, t_j) = p(t_j) \prod_{i=1}^{N} p(\mathbf{y}_i | t_j) \tag{2.26}$$

Equation 2.26 shows how the segmental HMM factors the distribution over the observations into intra-segmental and extra-segmental variation, as described in Section 1.3.3 on page 8. The extra-segmental distribution $p(t_j)$ models properties of the speech parameters which remain static over entire segments, such as speaker identity, while the intra-segmental variation $p(\mathbf{y}_1^N \,|\, t_j)$ characterises the distribution of the observations around a given target once external factors have been accounted for. This is distinct from standard HMMs where both variation sources are modelled together by the mixture components of the output distribution regardless of the time-scale over which they occur.

Static univariate Gaussians provide both inter-segmental and intra-segmental distributions in the original segmental HMM (Russell 1993). Gales & Young (1993) extended the model to include a mixture of Gaussians as the intra-segmental distribution. Later work reported in Holmes & Russell (1995) reverted to a uni-modal intra-segmental distribution, though used a dynamic target in which the slope and segmental mid-point value of a linear trajectory were drawn from univariate Gaussian distributions. Initial experimentation on a small corpus found the linear trajectory model to be a good estimator of the lower order Mel-scale cepstral coefficients, though short segments caused problems in estimation of both inter-segmental and extra-segmental distributions (Holmes 1996). Classification experiments on the TIMIT corpus found that the trajectory segmental HMMs outperformed both static segmental HMMs as well as standard HMMs using either monophone or biphone models, with or without feature derivatives. (Holmes & Russell 1999)

2.3.2 Segmental feature HMMs

Rather than directly modelling sequences of frames, another approach has been to model the features derived from a series of frames. Gish & Ng (1993) fitted a polynomial of the form

$$C = ZB + E \tag{2.27}$$

to each segment of length N.

$C = Y_1^N$ contains all the observations for a segment of N frames as row vectors, Z is an $N \times R$ design matrix, B an $R \times D$ trajectory parameter matrix, and E a matrix of residual errors. The order of the polynomial is governed by R, and Z chosen such that segment durations are normalised by altering the internal frame spacing according to the overall segment length. For example, for $R = 3$ and an N frame segment, Z becomes:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{N-1} & (\frac{1}{N-1})^2 \\ 1 & \frac{2}{N-1} & (\frac{2}{N-1})^2 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix}$$

and so

$$c_{n,i} = b_{1,i} + b_{2,i}\left(\frac{n-1}{N-1}\right) + b_{3,i}\left(\frac{n-1}{N-1}\right)^2 + e_{n,i} \tag{2.29}$$

where $n = 1, \ldots, N$ and $i = 1, \ldots, D$. For any given segment, polynomial model parameters are found using a least squares approach. Segment duration N, along with B and E are then used in place of the original features in an HMM system. With phones as segments, significant improvement over an HMM baseline system was demonstrated on a 20 keyword spotting task. However, the main drawback of such an approach is that phonetic alignments must be available or derived by some other means.

Yun & Oh (2002) developed this idea using the polynomial features as input to a segmental HMM, making full recognition practical rather than just classification of pre-aligned data. However, in their system, the parametric features are computed over fixed-length sliding windows of frames rather than variable-length time-normalised segments. In this case, with an analysis window of $N = 2M+1$ frames, C becomes $C_t = Y_{t-M}^{t+M}$, where the rows are vectors of features centred on $\mathbf{y}_t^T$. The design matrix is modified accordingly, so that the rows reflect relative positions from the current time. Again taking the case with $R = 3$ as an example, Z would be set to:

$$\begin{bmatrix} 1 & -\frac{M}{2M} & (-\frac{M}{2M})^2 \\ 1 & -\frac{M-1}{2M} & (-\frac{M-1}{2M})^2 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & \frac{M-1}{2M} & (\frac{M-1}{2M})^2 \\ 1 & \frac{M}{2M} & (\frac{M}{2M})^2 \end{bmatrix}$$

Context independent recognition on TIMIT was used to compare the new segmental feature HMM (SFHMM) with standard frame-based HMMs. Experimental results showed that the SFHMM gave a modest improvement on a standard HMM baseline using an MFCC parameterization of the acoustics with no δ features. However, an SFHMM system using MFCCs and their δs outperformed a standard HMM system which included both δ and $\delta\delta$ features. For systems with 2 Gaussian mixture components, the baseline HMM gave a recognition accuracy of 57.0%, and a SFHMM with fixed segment length of 5 and polynomial order 4 gave an accuracy of 60.1%. In this case, the segmental polynomial features were demonstrated to offer more than simply replacing the 2^{nd} order derivative coefficients.

2.3.3 Modelling temporal dependencies with HMMs

Appending δ and $\delta\delta$ parameters to feature vectors has become the standard approach by which a model of speech signal dynamics is included in state-of-the-art HMM speech recognition systems such as HTK (Young 1993). The properties of frame-based models of dynamic features are discussed in section 4.3 on page 102. Another approach which aims to account for the time dependencies present in the speech signal was taken by Woodland (1992). In this work, the mean vector of the output distribution associated with each state is modelled as being dependent on other observations. With the system occupying state $q_j \in \mathcal{Q}$, and y_t denoting the observation at time t, the predicted observation $\hat{y}_t^{q_j}$ is given by:

$$\hat{y}_t^{q_j} = \mu_0^{q_j} + \sum_p A_{k_p}^{q_j} (y_{t+k_p} - \mu_{k_p}^{q_j}) \tag{2.31}$$

In this equation, k_p is the amount by which the explanatory variable is offset from the current time, and $\mu_{k_p}^{q_j}$ and $A_{k_p}^{q_j}$ are the mean vector and prediction matrix associated with offset k_p respectively. This model reduces to a standard HMM when $A_{k_p}^{q_j}$ is set to be zero.

The model was evaluated on a multi-speaker isolated-word British English E-set[2] database for which 12 MFCCs and their corresponding δs were derived giving a 24-dimensional feature vector. Using a single predictor at offsets of $+3$ or -3 frames gave similar results, with test set errors of 3.9% and 3.8% respectively. These compare favourably with the baseline system result of 5.6% errors, in which full covariance Gaussian distributions were used to model the observations. However, when predictors at both $+3$ and -3 frames were used, the error rate increased to 6.9%. The suggested cause was that in this case there were more parameters than could be reliably estimated from the available data. A discriminative approach to training was then employed, and using a predictor with an offset of -3, the error rate was further reduced to 2.8%.

2.3.4 Modelling speech-signal trajectories with standard HMMs

Iyer et al. (1998, 1999) looked at a means of modelling the time dependencies in the parameterized speech signal without sacrificing any of the computational efficiency of a standard HMM system. They suggested that in fact, HMMs do model the trajectories in acoustic features. However, this does not rely on any knowledge of the underlying structure and instead is achieved through switching mixture component as necessary. They proposed modelling each phonetic unit with a set of M parallel HMMs, in which transitions are made left to right along individual HMMs, but not across the parallel paths. In an analogous fashion to segmental HMMs, intra-trajectory and extra-trajectory distributions are then accounted for independently.

Good initial parameter estimates were important to ensure that the inter-trajectory variation was captured. Otherwise, the model could simply degenerate into multiple HMMs for each phonetic unit. Initialisation first used a standard HMM to give a state-

[2]An E-set database is composed of the words 'B', 'C', 'D', 'E', 'G', 'P', 'T', and 'V'

level alignment. The observations corresponding to each state were then assigned to a particular path according to the clustering of a trajectory model. Two such models were experimented with, the first being the parametric feature model of Section 2.3.2, and the second was a non-parametric model described in Siu, Iyer, Gish & Quillen (1998). Once initialised, the parallel path HMMs were trained using the same techniques as standard HMMs.

Recognition results are reported on the Switchboard and Callhome corpora (Godfrey, Holliman & McDaniel 1992) with 2-path models. A slight decrease in word error rate was found when the models were used directly to produce output probabilities. However, when the parallel path HMM probabilities were combined with existing HMM scores in an output lattice, a 1% absolute, or 2.9% relative, error reduction was found.

2.3.5 ANNs in segment modelling

Hybrid ANN/HMM ASR occupies the territory between frame-based and segment models as the inputs are neither frames nor segments, but context windows over the observations. Artificial neural networks have been used to provide posterior probabilities for standard HMMs, segment models, as well as the observation process in non-linear state-space models.

ANN segment models

Zavaliagkos, Zhao, Schwartz & Makhoul (1994) describe a hybrid segmental ANN/HMM model which was applied to word recognition on the RM corpus within an N-best paradigm. Experiments used single layer or elliptical basis networks to rescore word lattices generated using standard HMMs. Artificial neural networks require a fixed-sized input layer, and two methods of time normalisation were compared. The first used a quasi-linear sampling of the feature vectors, either repeating or ignoring frames of features, but never actually interpolating. The second took a discrete cosine transformation (DCT) of the sequence of feature values across each dimension of the observations. These were truncated to give a fixed number of coefficients. The DCT approach was found to give the best final performance. For 5 RM test sets, lattice rescoring using a combination of both types of network and HMMs gave error reductions of between 9% and 29% relative to an HMM baseline.

Verhasselt, Cremelie & Marten (1998) worked with a similar idea, using ANNs to give posterior probabilities for entire segments. However, the single layer networks were replaced with multi-layer perceptrons (MLP) and a pre-segmentation algorithm was included which limited the set of allowable segmentations, making full recognition feasible. Another addition was that the probabilistic framework made explicit use of segmentation probabilities. Phone recognition on the TIMIT core test set combined the outputs of diphone and context independent phone segment scores, giving a recognition accuracy of 70.2%, comparable to some of the highest reported performances for this task.

Non-linear state-space modelling of speech

Richards and Bridle (1999) introduced the hidden dynamic model (HDM) which sought to give an explicit model of co-articulation within a segmental framework. A static target or series of targets in a hidden state space was associated with each phone in the inventory. Given a time-aligned sequence of phones, a Kalman smoother was run over the targets to produce a continuous trajectory through the state space. These trajectories were connected to the surface acoustics by means of a single MLP mapping.

All targets were set to zero and network weights randomised at the start of training, which was by a form of gradient descent. Preliminary results are reported in Picone, Pike, Regan, Kamm, Bridle, Deng, Ma, Richards & Schuster (1999) for an N-best rescoring task on the switchboard corpus. With a baseline word error rate of 48.2% from a standard HMM system, 5-best rescoring with the reference transcription included using the HDM gave a reduced error rate of 34.7%. An identical rescoring experiment using an HMM trained on the data used to build the HDM gave a word error rate of 44.8%. This suggests that the HDM was able to capture information that the HMM could not.

Iso (1993) investigated a similar model in which a set of observations $\mathbf{y}_1, \dots, \mathbf{y}_t, \dots, \mathbf{y}_N$ were connected to a sequence of control commands $\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_K$ by means of a non-linear predictor:

$$\hat{\mathbf{y}}_t = f(\mathbf{y}_{t-1}, \mathbf{x}_k) \tag{2.32}$$

As with the HDM, the control commands were static targets, though diagonal covariance Gaussian distributions over the state-space were used in place of state vectors. Parameters were estimated using a form of gradient ascent, and two alternative objective functions were suggested, one maximum likelihood and the other discriminative. Some preliminary experiments confirmed the benefit of diagonal over identity covariances for the control commands, and also improved results using a discriminative approach to training. However, no baseline using a conventional system was given so real benefits are hard to quantify.

2.3.6 Stochastic segment models

In the late 80's, Mari Ostendorf and members of her group in Boston began an investigation into segment modelling. They introduced the stochastic segment model (SSM)[3], a framework within which to research the distributional forms and modelling assumptions necessary to account for inter-frame correlations and temporal dependencies.

Treating the segment's length $l \in \mathcal{L}$ as a random variable, the stochastic segment model describes a set of p-dimensional observations $\mathbf{y}_1^l = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$ using a joint Gaussian distribution. Model m generates the observations $\mathbf{y}_1^l$ according to the density

$$p(\mathbf{y}_1^l, l|m) = p(\mathbf{y}_1^l \,|\, l, m)\, p(l|m) \tag{2.33}$$

where the acoustic and duration models provide $p(\mathbf{y}_1^l|l, m)$ and $p(l\,|m)$ respectively. Rather than describing $\mathbf{y}_1^l$ directly, the original formulation of the SSM modelled fixed-length (see page 33) segments denoted $\mathbf{z}_1^N = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. An observed segment $\mathbf{y}_1^l$ is considered to be a linear down-sampling of $\mathbf{z}_1^N$ using the relation $\mathbf{y}_1^l = \mathbf{z}_1^N T_{m,l}$ in which the original feature dimension is preserved. Writing $\mathcal{Y} = \mathbf{y}_1^l$ and $\mathcal{Z} = \mathbf{z}_1^N$, the acoustic model likelihood $p(\mathcal{Y}|l, m)$ is then given as the following marginal distribution:

$$p(\mathcal{Y}|l, m) = \int_{\mathcal{Z}:\mathcal{Y}=\mathcal{Z}T_{m,l}} p(\mathcal{Z}|m)\, d\mathcal{Z} \tag{2.34}$$

[3]The reader who is familiar with the original SSM literature should be aware that some of the notation has been altered to fit in with the conventions used in this thesis.

Modelling assumptions used in the SSM framework

In the most general case, known as **full covariance**, $\mathbf{z}_1^N$ was modelled using an Np-dimensional Gaussian. This takes full account of the intra-segmental correlation structure, though requires a large number of parameters. Modelling a p-dimensional observation with N frames would require a $Np \times Np$ covariance matrix. Robust estimation of such a model would require a great deal of data, and was in fact found to be impractical for systems with more than a few feature dimensions (Digalakis, Ostendorf & Rohlicek 1989).

The **independent-frame**, or block-diagonal model reduces the parameterization by assuming that successive frames are independent given l, the length of the segment. The probability of the fixed-length vector sequence $\mathbf{z}_1^N$ under model m is then:

$$p(\mathbf{z}_1^N|m) = p_1(\mathbf{z}_1|m)p_2(\mathbf{z}_2|m)\dots p_N(\mathbf{z}_N|m) \tag{2.35}$$

where p_i is the probability density which models the i^{th} frame. Even though the inter-frame correlation modelling, the potential advantage of SSMs, is ignored under such a model, it has been shown to match HMM performance. Digalakis et al. (1989) report that with similar number of parameters and context-independent models, an HMM and an independent-frame SSM produce the same phone classification accuracies.

A **target-state** model assumes that every segment model is associated with a static target in some state-space, $\mathcal{X}$. The target might vary according to factors such as context or speaker characteristics, and is modelled by a zero-mean Gaussian distribution:

$$\mathbf{z}_t = H_t\,\mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{2.36}$$

$$\mathbf{x}_t = \boldsymbol{\eta}_t \qquad \boldsymbol{\eta}_t \sim N(\mathbf{0}, D) \tag{2.37}$$

Individual frames are independent when conditioned on the state target.

A **Gauss-Markov** structure provides a means of loosening the framewise independence assumption without causing a major increase in the number of parameters. Under a model of this type, each frame is dependent only on the last, i.e.

$$p(\mathbf{z}_1^N) = p_1(\mathbf{z}_1)\prod_{i=2}^{N} p_i(\mathbf{z}_i|\mathbf{z}_{i-1}) \tag{2.38}$$

where the $p_i(\mathbf{z}_i|\mathbf{z}_{i-1})$ are conditional Gaussian densities. As shown by Digalakis (1992), and in Section 3.4, most of the variation in a given frame of acoustic speech data can be explained using the previous frame as a predictor, which supports this choice of model structure.

For both the target-state and Gauss-Markov models, TIMIT context independent classification accuracies were higher, 52.0% and 53.9% respectively, than for the independent-frame model, which gave an accuracy of 50.6% using a feature set of 18 MFCCs (Ostendorf & Digalakis 1991, Digalakis 1992). When δ coefficients were included in the feature set, performance using the target-state and Gauss-Markov models deteriorated, giving lower classification accuracies than an independent frame model. This result was attributed to non-linearities near segment boundaries. Furthermore, the lack of an observation noise term and the associated smoothing in the Gauss-Markov model was thought to contribute to a mismatch between training and test data (Digalakis 1992).

A generalisation of the Gauss-Markov model is the **dynamic system** (DS) model, referred to in this thesis as the linear dynamic model (LDM). These were introduced in Section 2.1.2 on page 20 and are described fully in Chapter 4.

Dynamic system models a.k.a. LDMs

These models were the focus of Vasilios Digalakis' thesis (1992), where approaches to estimation, classification, and recognition with the models were presented. Digalakis termed the assumptions made in modelling fixed-length and variable-length segments with LDMs *trajectory invariance* and *correlation invariance* respectively. *Trajectory invariance* assumed that there was a single, fixed trajectory underlying all instances of a phone. In the standard SSM fashion, a linear time-warping was used to normalise segment durations before the model was applied. *Correlation invariance* assumed that within each segment, there were a number of regions in which inter-frame correlations were static. In this case, no time-warping was used, but a deterministic mapping dictated the durations of each of the sub-segmental models.

On a TIMIT classification task, with phones as segments, similar results were found under each of the two assumptions for longer phones. However, poor performance arose using the trajectory-invariance formulation when an observation sequence was considerably shorter than its hypothesised length. The problem was thought to be that under the trajectory invariance assumption, the correlations originally present in short instances of segments were reduced after duration normalisation. Correlation-invariance was adopted for all further work with the model.

Linear dynamic models were shown to outperform all the SSM models above on TIMIT classification, the highest accuracy being 73.9% after the 61 phone models were folded down to 39 in the standard way as described on page 150. The closest performance by a static SSM, the independent frame model, was an accuracy of 72.1% for an identical task.

Recognition was implemented using an iterative 'split-and-merge' algorithm. An initial alignment is given, and then at each iteration, all hypothesised phone segments can be split in two at the middle frame or merged with a neighbouring segment. The new segmentation which gives the highest likelihood is chosen and the procedure is repeated until no increase in the log-likelihood is found. Details of the basic algorithm, along with modifications and the theory of local search algorithms can be found in Digalakis (1992).

With a uniform initial segmentation, the LDM system gave a TIMIT phone-recognition accuracy of 57%, higher than the 55% found using independent-frame models. This accuracy was increased to 63% on inclusion of δ features, though in this case an independent frame model gave a slightly higher 64%. These results suggest that under this implementation, the advantage of modelling the underlying dynamics with LDMs was lost with the addition of derivative information. Note that these experiments use an early release of the TIMIT database with a different allocation of training and testing data to that which later became standard. This precludes direct comparison with results given in this thesis using similar models.

Segmental mixtures

Kimball (1994) developed some of the ideas of the stochastic segment model, considering the addition of mixture modelling. Gaussian mixtures rather than single Gaussians were used for frame-level modelling in the independent-frame SSM. This gives a model which can be cast as an HMM with a deterministic state sequence dependent on the segment length. The model was further augmented to have segment-level mixtures. In this case the model can be derived as a segmental mixture HMM as described by Gales & Young (1993), though with a fixed state transition sequence dependent on the segment length. Recognition experiments on the RM corpus showed similar results for these two systems, both of which gave a lower word error rate than an independent-frame SSM baseline.

2.3.7 Goldenthal's statistical trajectory models

Goldenthal (1994) also explored methods for explicitly modelling the dynamics underlying speech sounds. His approach involved modelling each speech segment using a *track* consisting of a sequence of averaged acoustic parameters. Variable-length and fixed-length segments were experimented with, and the fixed, or *trajectory invariant* tracks adopted. A form of fractional linear interpolation was used for the time normalisation in which each acoustic observation contributed its data proportionally to adjacent frames of the track according to how closely the time-scales corresponded. The tracks were typically set to a length of 10 frames.

Just as with the full covariance SSM of Section 2.3.6, data limitations meant that estimation of a full covariance matrix for each track was not practical. One approach to reducing the number of parameters would be to split each model into regions in which the full correlation structure is modelled. However, in this work, residuals were first calculated by differencing corresponding frames of track and observations. These errors were then partitioned into Q regions in which the full correlation structure was modelled. If a model has accounted for the dynamics and correlation structure of a segment, then the residual energy will be small and show weak inter-frame dependencies in comparison to the original data. It was proposed that by partitioning at this level, little information is sacrificed. Sub-segmenting the residual tracks into $Q = 3$ regions gave a context-independent TIMIT classification accuracy of 74.2%, and then 75.2% with the log-duration of the token modelled as a Gaussian and built into the error distribution.

Two approaches to dealing with contextual variation were also investigated. The first involved creating clusters of biphone tracks covering each model in each left and right context independently, which could then be merged to create any given triphone. The second made explicit models of the dynamics at segment boundaries by generating tracks for every transition in the training data. These were clustered to provide 200 transition models which were incorporated in the acoustic likelihood calculation during recognition.

Context-independent phone recognition on the TIMIT core test set gave an accuracy of 61.9%, where the usual confusions as given on page 150 were allowed in reporting results. This was increased to 63.9% when the transition models were included. Similarly, the transition models improved context-dependent recognition performance, with the accuracy of 66.5% being raised to 69.3%. These results compare favourably with the state-of-the-art in TIMIT phone recognition which will be summarised below, though it should be noted that trigram language models were used in producing the context-dependent results, rather than bigrams which are normally used when reporting TIMIT recognition results.

2.4 Conclusions

In this chapter we have outlined non-standard approaches to acoustic modelling for speech recognition. The models of Section 2.2 incorporated articulatory information, and in Section 2.3, the models sought to account for the temporal dependencies present in speech data. Many of these techniques were shown to be useful compared to acoustic-only or frame-based baseline models, however it is interesting to compare these results with others in the literature.

Many of the experiments described above give results for TIMIT phone recognition, a well known task which is commonly used for development of ASR systems. Currently, the highest reported accuracy is by Robinson (1994) using a hybrid ANN/HMM system. Chen & Jamieson (1996) report a result which may be higher, though a percent correct is quoted rather than recognition accuracy. This is an unreliable measure since it does not penalise insertion errors.

Table 2.2 shows the recognition accuracies for the systems described in this chapter, along with the best reported ANN/HMM and HMM results for this task. In all cases apart from Goldenthal's context-dependent models, bigram language models were used. All results are on the NIST core test set which contains the 8 `si` and `sx` sentences from 2 male and 1 female speakers from each of the 8 dialect regions. Results on the core test set tend to be slightly lower than those on the full test data, which may be due to the balance of dialect and gender not following that of the training data. Note that Table 2.2 does not include Yun & Oh (2002) as results were based on the full test set and also

Author	phone recognition accuracy
Goldenthal, CI (Section 2.3.7)	64.9%
Goldenthal, CD (Section 2.3.7)	69.5%
Verhasselt, CI + diphone (Section 2.3.5)	70.2%
Deng (Section 2.2.2)	73.0%
Lamel & Gauvain (1993*a*), CD triphone HMM	69.1%
Deng baseline, CD triphone HMM (Section 2.2.2)	70.9%
Robinson (1994), CI hybrid ANN/HMM	73.4%

Table 2.2: Summary of TIMIT phone recognition accuracies for the systems described in this chapter, along with the highest reported ANN/HMM and HMM accuracies for this task. Context dependent and context independent models are denoted by CD and CI respectively

included the `sa` sentences, which are the same for every speaker in both training and test sets. The work of Digalakis (1992) is also excluded as experiments use an early version of TIMIT which follow a different allocation of training and test speakers.

However, it is apparent that many of these methods give accuracies which are close to the state-of-the-art. Indeed, Deng's articulatory-motivated state selection gives the highest HMM accuracy, and both Verhasselt and Goldenthal produce systems which outperform a standard triphone HMM recognizer.

Chapter 3

Preliminaries

This chapter will give a review of the data and some of the techniques which will be used in later experimentation. Section 3.4 then gives some preliminary analysis which examines the suitability or otherwise of linear models of speech data.

3.1 Data collection and processing

Both acoustic and articulatory data is used in this thesis. The corpora which will be used for experimentation are described below, however this will be preceded by an outline of the issues involved in collecting and processing data from these distinct information sources.

3.1.1 Articulatory Data

Incorporating articulatory features into speech recognition has already been stated as one of the concerns of this work. This of course relies on having access to measured or automatically generated articulation.

Direct measurement of human articulation

Measuring articulation is frequently an invasive process. However, subjects generally manage to produce intelligible and reasonably natural speech despite the array of instruments to which they are attached. The most commonly used devices are:

X-ray microbeam An x-ray microbeam system involves attaching 2-3mm gold pellets to the chosen set of articulators. A narrow high-energy x-ray beam then tracks the pellets during speech production. The output is a series of samples of the x and y coordinates of each articulator in the mid-sagittal plane. The system described by Westbury (1994) uses sampling rates of between 40Hz and 160Hz dependent on the particular articulator which is being tracked. Rather than subjecting the entire head to radiation, a focused beam only scans the areas where the pellets are expected to be, predicting at each sample where to scan at the next. Figure 3.1 shows a schematic diagram of such a system.

One drawback of x-ray microbeam measurement of articulation is that the machinery required can produce appreciable levels of background noise. Not only does this result in a noisy signal, but can also interfere with speech production. This is due to the Lombard effect, which describes the reflex by which a speaker modifies their vocal effort whilst speaking in noisy surroundings (Junqua 1993).

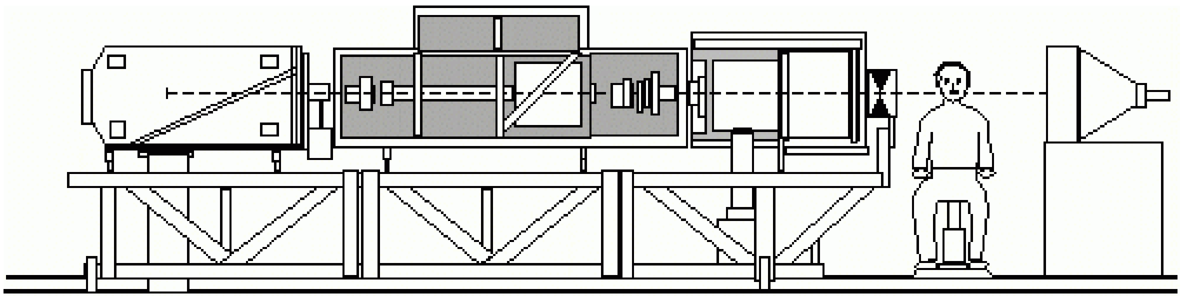

Figure 3.1: Schematic diagram of an x-ray microbeam system for measuring human articulation. Diagram reproduced with permission from Dr. Michiko Hashi's homepage, found at `http://www.hoku-iryo-u.ac.jp/ mhashi/index.html`

EMA An Electromagnetic Articulograph (EMA) system requires fixing small receiver coils to the chosen articulators. Alternating magnetic fields are produced by fixed transmitter coils mounted on a helmet. Each receiver coil is wired to a circuit which measures the induced current, from which the distance between transmitter and receiver is calculated. As with an x-ray microbeam system, the x and y coordinates of each sensor in the mid-sagittal plane are sampled over time. Figure 3.2 shows the placement of coils which was used in collecting the MOCHA (Wrench 2001) EMA database, detailed below in Section 3.2.1

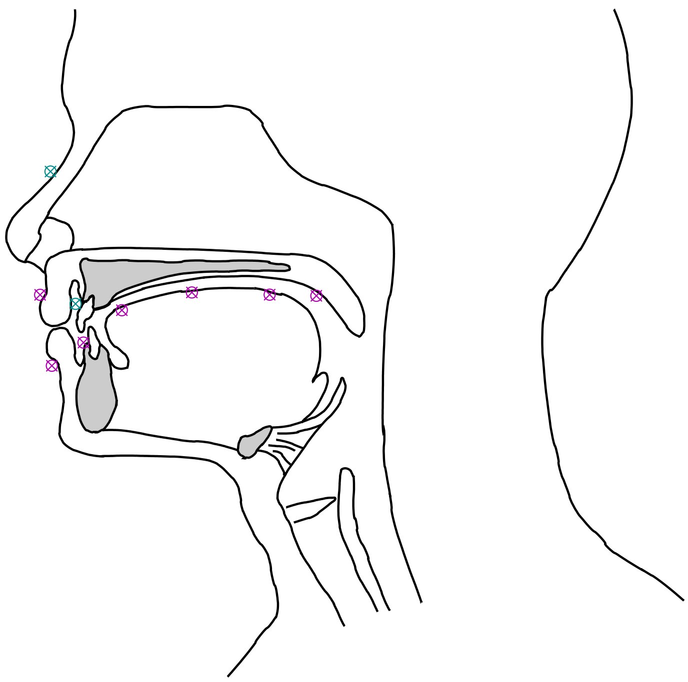

Figure 3.2: Placement of coils which was used in recording the MOCHA EMA database. Coils marked in magenta are attached to articulators, and regular samples of their position in the mid-sagittal plane taken. These correspond to tongue tip, body and dorsum, lower incisor, velum and upper and lower lip. The coils marked in cyan are included to provide reference points which allow head movements relative to transmitter coils to be corrected for. Diagram reproduced from Richmond (2001a) with permission of the author.

Laryngograph The laryngograph measures the variation in conductance between transmitting and receiving electrodes positioned either side of the larynx. This variation is related to the change in glottal contact. Sampling is generally at the same rate as used for the audio signal, producing a waveform from which pitch and voicing information can be derived.

Electropalatograph The electropalatograph (EPG) measures tongue/palate contact over the whole palate. The user wears an artificial palate which has a number of electrodes embedded on the lower surface. A grounding electrode is attached to the speaker and a series of recordings of the points at which the tongue is in contact with the palate is made.

Corpora large enough for training and evaluating ASR systems are rare due to the expense and labour intensity of data collection. However, two such data-sets exist: MOCHA (Wrench 2001) which will be described on page 58, and the Wisconsin x-ray microbeam database (Westbury 1994). The latter was introduced on page 25 of the literature review and consists of parallel articulatory and acoustic features for 60+ subjects each of whom provide in the region of 20 minutes of speech data. The data for each subject is divided into a number of tasks, which include reading prose passages, counting and digit sequences, oral motor tasks, citation words, near-words, sounds and sound sequences along with read sentences.

Automatically recovering articulatory parameters

Building a mapping from acoustic features to the articulatory gestures which produced them, known to speech researchers as the *inversion mapping*, is by no means trivial. Firstly, the mapping is bound to be highly non-linear, since in some instances, small shifts in the articulators produce significant modification of the acoustic signal. For example, consider the effect on the acoustic signal of opening the lips during a plosive. Here, a small movement results in a significant change in pressure within the vocal tract. The result is a sudden change in air-flow, and therefore also the acoustic signal. Secondly, the mapping is an example of an ill-posed problem, as there is no one-to-one mapping from the acoustic to articulatory domains.

This is demonstrated empirically by Roweis (1999), who investigated the geometric spaces articulatory and acoustic parameters occupy, and how they relate. It was shown that articulatory configurations can be mapped to points in acoustic feature space using simple linear transformations, though the reverse was not so. An experiment using the parallel acoustic-articulatory data collected at the Wisconsin x-ray microbeam facility (Westbury 1994) was performed to demonstrate this many-to-one relationship. With the acoustic signal parameterized using line spectral pairs (LSP), a key acoustic frame was chosen. The 1000 frames nearest the key frame in acoustic space were plotted in articulatory space. The entire database was used, which meant that there was no distinction made between effects due to intra-speaker and inter-speaker variation. The spread of points in articulatory space found for acoustically similar features was large compared to reference ellipses showing the 2 standard deviation contours for the 1000 frames which were closest in articulatory space. Many of the plots also show multimodality in the spread of the points in articulatory space. These findings demonstrate that a range of articulatory configurations can be used to produce acoustically similar sounds.

The phenomenon of articulatory compensation, which occurs when someone is speaking under unusual constraints, can be used to demonstrate the one-to-many nature of the acoustic-articulatory mapping on a speaker-dependent basis. Such a constraint can be artificially introduced in the form of a bite-block, which holds the speaker's jaw in an unnatural position. Lindblom, Lubker & Gay (1979) recorded subjects saying four Swedish vowels both with and without bite-blocks in place. The finding was that all subjects were able to produce formant patterns within their usual range of variation despite the constraint. In fact, anyone who has seen a ventriloquist perform knowns that speech can be produced with little or no visible movement of the articulators.

Critical articulators Articulatory inversion, it seems, is going to prove challenging, even for normal, unconstrained speech. However, for the purposes of speech recognition, it may be simplified by not requiring faithful recovery of the position of all articulators at each time. An articulator which has a fundamental role in the production of a phone is said to be *critical* for that phone. For example, the behaviour of the lips and velum are critical in the production of a [p], whereas the motion of the tongue is far less important. If for a given phone, a subset of the articulatory features provide the crucial discriminatory cues, accurately recovering those features may be the goal of an inversion mapping for ASR. Papcun, Hochberg, Thomas, Laroche, Zachs & Levy (1992) reported that 'critical articulators are less variable in their movements than non-critical articulators', a finding that has been supported in work on acoustic-articulatory inversion. Richmond, King & Taylor (2003) observed that the most confident (i.e. those with the lowest associated variance) predictions of articulatory position using a univariate mixture density network (MDN) were generally for articulators which were considered critical for the production of a given phone.

The difficulties inherent in accurate inversion mapping have not served to deter the large number of researchers who have worked on this problem. In recent years, the development of articulography technologies such as x-ray microbeam and EMA, along with increases in computing power, have opened articulatory inversion to machine learning techniques. Approaches found in the literature include artificial neural networks, Kalman smoothers, self organising HMMs and use of codebooks. A review of such work can be found in Richmond (2001).

3.1.2 Acoustic Data

The source-filter model of the vocal tract provides the assumptions underlying many common approaches to speech processing. These include filter banks, cepstral analysis and linear predictive coding (LPC). Analysis generally assumes that the speech signal is statistically stationary over short time intervals, or frames. Within one such frame, the source-filter model (with ω denoting frequency) decomposes the spectrum of the speech signal into an excitation $E(\omega)$ and a vocal tract transfer function $V(\omega)$. This relationship is written as:

$$X(\omega) = E(\omega)\,V(\omega) \tag{3.1}$$

During spoken English, two main sources of acoustic energy are used to excite the filter. The first is due to vibrations of the glottis which produce a periodic airflow waveform, such as occurs during voiced speech. The second is when air expelled from the lungs is passed through a narrow constriction creating the turbulence which characterises frication.

The acoustic signal is an encoding of the combination of an energy source and the spectral modifications imparted by the vocal tract transfer function. The information which it conveys is not immediately apparent, and so features must be extracted. It is the vocal tract shape, here modelled by the filter, which creates resonances at certain frequencies. Since these resonances cause the acoustic signal to carry the frequency patterns which distinguish speech sounds, automatic speech recognition includes a step in which filter information is extracted from the acoustic signal prior to any subsequent modelling.

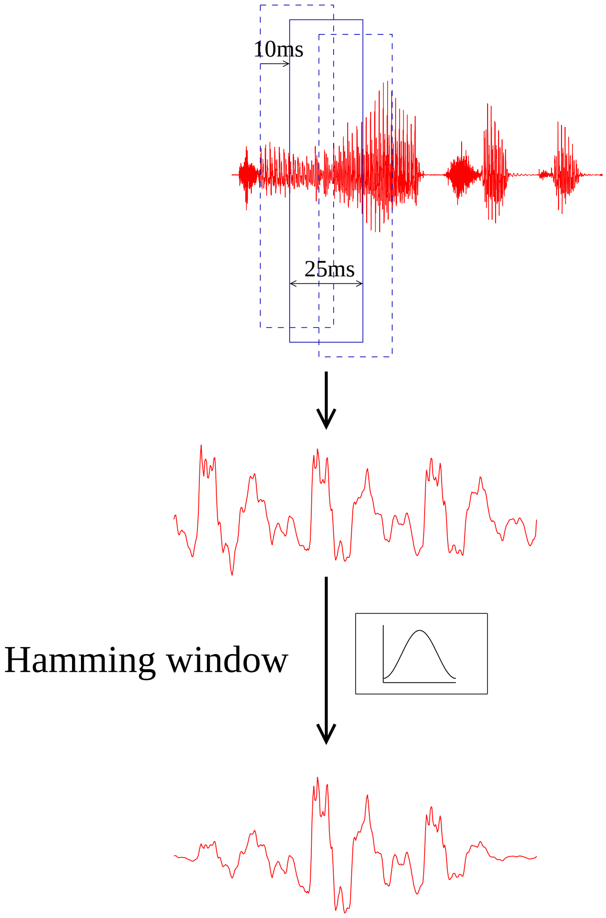

Figure 3.3: Spectral estimates are made within a series of regularly spaced overlapping windows. Hamming windows each spanning 25ms placed in the region of 10ms apart are commonly used.

As mentioned above, feature extraction assumes the acoustic signal to be statistically stationary over short regions during which estimates of the spectrum are made. Figure 3.3 shows the process of windowing the acoustic signal prior to analysis. There is a trade-off between frequency and time resolution which must be made: longer analysis windows allow more detailed estimates of the frequency components of the signal, though have the effect of smearing spectral characteristics in time.

Spectral estimates are made within a series of regularly spaced overlapping windows, with a common choice being a set of Hamming windows each spanning 25ms placed in the region of 10ms apart. Mel-frequency cepstral coefficients (MFCCs) and perceptual linear prediction (PLP) cepstra are currently the most widely used features for ASR. Variants exist, such as RASTA-PLP (Hermansky, Morgan, Bayya & Kohn 1991) which is tailored to ASR in noisy conditions. Producing MFCC and PLP parameterizations of the acoustic signal combine steps which are motivated by both speech production and perception, and others which tailor the parameters to the models which will be used to characterise them.

Investigating new front-end acoustic parameterizations does not form part of the work in this thesis. However, MFCCs and PLP cepstra offer distinct representations of the speech signal, one of which may be a more appropriate choice for the class of models under investigation. Therefore, experimental work will compare results found using both MFCC and PLP-based features.

Calculating MFCCs Calculating Mel-frequency cepstral coefficients first involves computing estimates of the spectrum with a fast Fourier transform (FFT). This is usually performed within tapered windows onto the acoustics, such as the Hamming windows of Figure 3.3. These estimates are then smoothed by integrating within a set of overlapping triangular filters, such as shown in the second box of Figure 3.4. The filters are spaced along a Mel-warped frequency scale which is roughly linear up to 1 kHz and logarithmic thereafter. Arranging the filter bank along such a scale has the effect of widening pass bands and the spacing of filters in the higher frequency ranges. The Mel warping provides the 'M' in MFCC, and is based on pitch perception experiments which demonstrate that in the low frequency ranges, the human auditory system can make distinctions between frequencies which are closer together than is possible in the higher ranges.

The filter bank coefficients are then subject to log compression. Applying a logarithmic scaling of the filter bank outputs transforms the model of Equation 3.1 from being multiplicative to additive, making the frequency components of $E(\omega)$ and $V(\omega)$ linearly separable. An inverse discrete cosine transform (IDCT) is applied to give the *cepstrum* of the speech signal. The spectral envelope $V(\omega)$, which varies slowly with respect to frequency, is represented by the first few coefficients, and the fine detail $E(\omega)$ by those of higher order. Truncation is used to give a smoothed estimate of the filter characteristics, where typically the first 12 coefficients are retained.

The IDCT has the effect of reducing, though not removing entirely, correlation between coefficients, thus simplifying subsequent modelling (Macho, Nadeu, Jancovic & Hernando

1999).

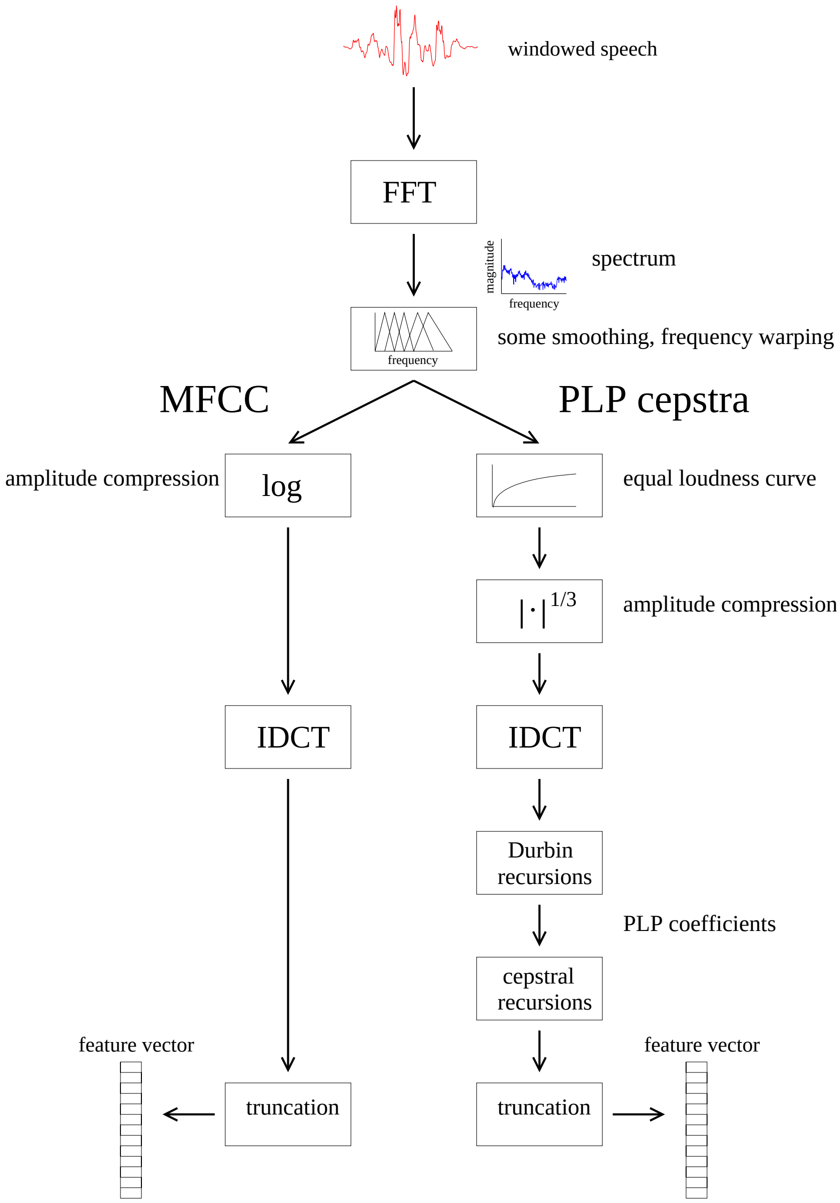

Figure 3.4: Figure showing the steps taken to compute Mel frequency and perceptual linear prediction cepstra. Adapted from figures in Young (1995) and Gold & Morgan (1999).

Calculating PLPs Figure 3.4 shows the steps involved in extracting PLP cepstra from the speech signal alongside those for deriving MFCCs. Producing these two parameterizations can be seen to involve many of the same steps. As for MFCCs, spectral estimates are made within a set of overlapping tapered windows onto the acoustic signal. These estimates are then smoothed by integrating within a set of overlapping triangular filters positioned along a Mel-warped frequency scale.[1]

An equal-loudness curve is used to weight the filter bank outputs to follow the variation in the sensitivity of human hearing across the frequency ranges, and amplitude compression uses a cube-root to approximate the intensity-loudness power law. An inverse discrete cosine transformation is applied to generate the real cepstrum, though unlike computing MFCCs where smoothing is achieved through truncation, the parameters of an all-pole filter such as used in standard linear predictive coding (LPC) are estimated. Finally, a set of cepstra can be computed from the LPC coefficients to give a parameterization in which the spatial correlations are reduced.

Computing MFCCs and PLP cepstra give sets of acoustic features with similar properties. The essential difference is that PLPs employ an extra layer of LPC-based smoothing. A further stage which is frequently taken in either case is liftering. The cepstra are reweighted to give emphasis to the higher order coefficients, which tend to have small magnitudes in comparison to those of lower order. For example, scaling the n^{th} coefficient by n has the effect of roughly equalising the variances of the cepstra (Gold & Morgan 1999).

[1] Some authors, such as Gold & Morgan (1999) suggest that spectral smoothing be implemented within a set of trapezoidal filters spaced along a Bark-warped frequency scale in calculating PLPs. However, HTK (Young et al. 2002), which uses triangular filters positioned along a Mel-warped scale, has been used for feature extraction in this work.

3.2 Corpora used in this thesis

The data which will be used in experimentation in this thesis comes from two corpora, MOCHA (Wrench 2001) and TIMIT (Lamel, Kassel & Seneff 1986). These, along with the sets of features which will be derived from them are described in the following sections.

3.2.1 MOCHA

The MOCHA database was recorded at Queen Margaret University, Edinburgh, and consists of parallel acoustic-articulatory information for a number of speakers, each of whom read up to 460 sentences. The sentences comprise the 450 American TIMIT sentences which were designed to provide good phone pair coverage, along with an extra 10 sentences which include phonetic pairs and contexts found in the received pronunciation (RP) accent of British English.

The MOCHA corpus includes automatically generated labels for the 46-phone set detailed in Appendix A.1. Once phone sequences had been generated for each utterance using a keyword dictionary (Fitt & Isard 1999), HTK (Young 1993) was used to force-align flat-start monophone HMMs to the acoustic data to give start and end times for each phone label. All experimental work uses the 20 minutes of speech data from the southern English female subject `fsew0` for which Wrench & Richmond (2000) estimate that 5-10% of phones are incorrectly labelled.

Feature sets

The MOCHA corpus offers a number of different parallel streams of acoustic and articulatory data from which a variety of feature sets can be derived. These comprise:

Acoustic The speech waveform was recorded directly onto disk in a sound-damped studio, sampled at 16kHz and stored with 16 bit precision. For this work, both MFCCs and PLP cepstra were generated from the acoustic signal at 10ms intervals within overlapping 25ms Hamming windows using the HTK version 3.1 tool `HCopy`. In each case, the resulting 12 cepstral coefficients were augmented to include the log signal energy along with δs and $\delta\delta$s corresponding to each of the parameters, giving a $(12 + 1) \times 3 = 39$-dimensional feature vector.

Articulatory The articulatory information recorded in the MOCHA corpus includes electromagnetic articulograph (EMA), laryngograph (LAR), and electropalatograph (EPG) data. The EMA data consists of samples of 14 articulatory dimensions recorded every 2ms. These correspond to x and y coordinates in the mid-sagittal plane for 7 points on the articulators: tongue tip, body and dorsum, lower incisor, velum and upper and lower lip. Figure 3.5 shows the EMA data and acoustic signal for the first sentence in the MOCHA `fsew0` data – 'This was easy for us.'.

Plotting the mean of each of the articulatory dimensions for each of the 460 utterances reveals a slight drift in average location throughout the corpus. The source of this systematic fluctuation is unclear, and possible causes might be processing error, temperature changes in the recording booth, or the subject changing speaking style as they become accustomed to the coils attached to their articulators. To remedy this, the means of each utterance were taken, ordered as they had been recorded and then low-pass filtered to extract the underlying trend. The articulatory variables were then normalised to lie in the range $[-1, 1]$ using the filtered values as utterance-specific means[2]. This normalised data was used in all experiments involving EMA data.

The laryngograph waveform shows change in glottal contact and is sampled at 16kHz. During voiced speech the vocal folds vibrate, and so have higher velocity than during unvoiced speech when they are at rest. This data is used to produce a measure of voicing energy every 0.01ms through differentiation of the signal and calculation of the root-mean-square of non-overlapping 160-sample windows.

The electropalatograph (EPG) provides information on the contact between tongue and palate as 62 on/off values sampled at 200Hz. The collector of the MOCHA data has applied principal components analysis (PCA) to reduce this data to a feature vector of 4 smoothly varying components sampled at 100Hz (Wrench & Hardcastle 2000).

[2]This post-processing was carried out by Korin Richmond.

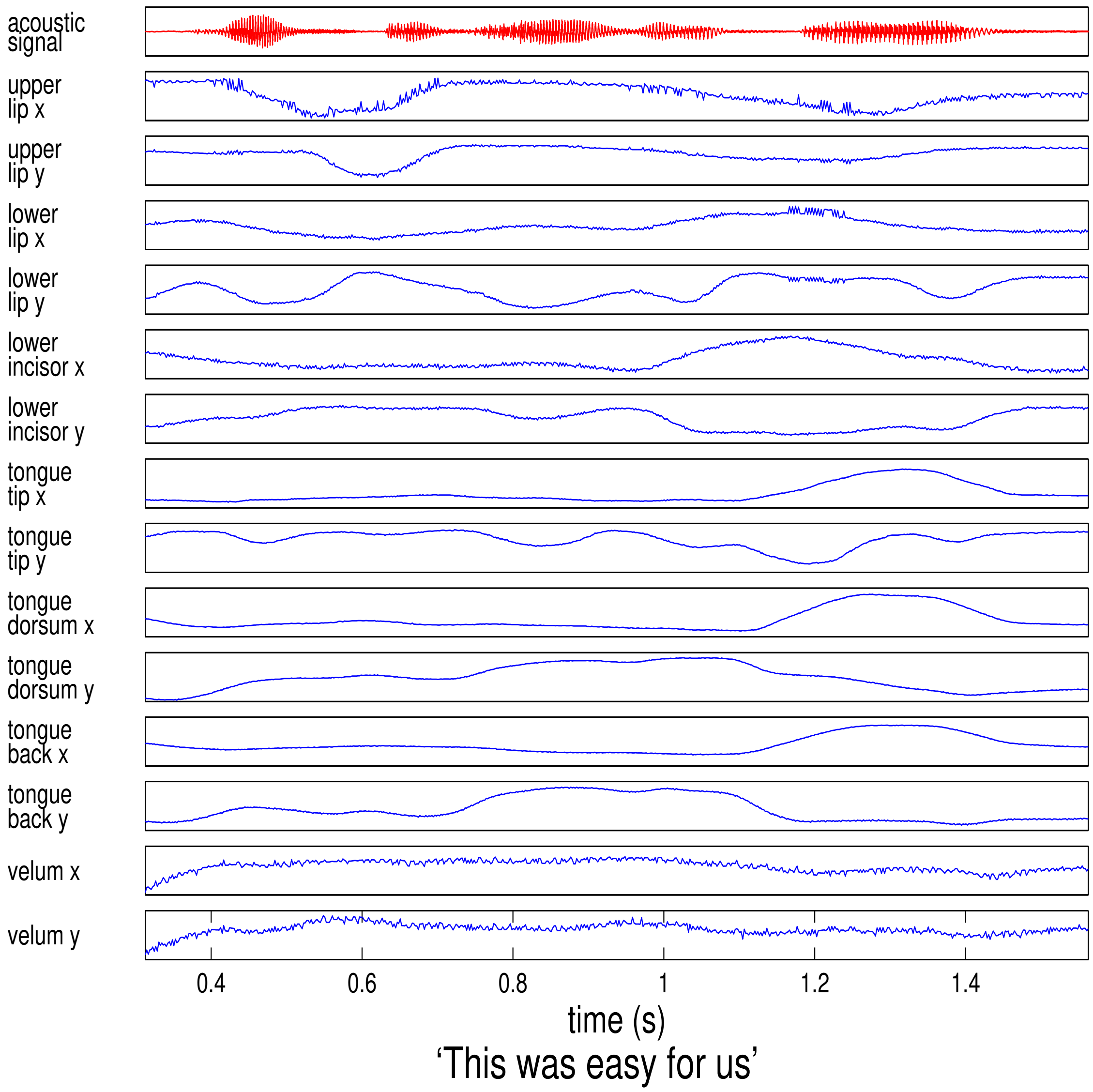

Figure 3.5: Figure showing the 14 articulatory dimensions contained in the EMA data, along with the acoustic signal for the first sentence in the MOCHA `fsew0` data – 'This was easy for us.'

Automatically recovered EMA The automatically recovered articulatory traces used in this work were estimated by Korin Richmond using a feed-forward neural network of the type shown in Figure 3.6. A set of 20 filter bank coefficients was computed within 20ms windows on the acoustic signal at 10ms intervals. A context window of 20 such frames provided the network's input layer of 400 units. The inputs were fully connected to a single hidden layer which in turn fed forward to the 14-dimensional articulatory output. The 14 dimensions correspond to x and y coordinates of the 7 points on the articulators present in the MOCHA EMA data. This data will be referred to as network-recovered EMA (net EMA). Further details can be found in Frankel et al. (2000) and Richmond et al. (2003).

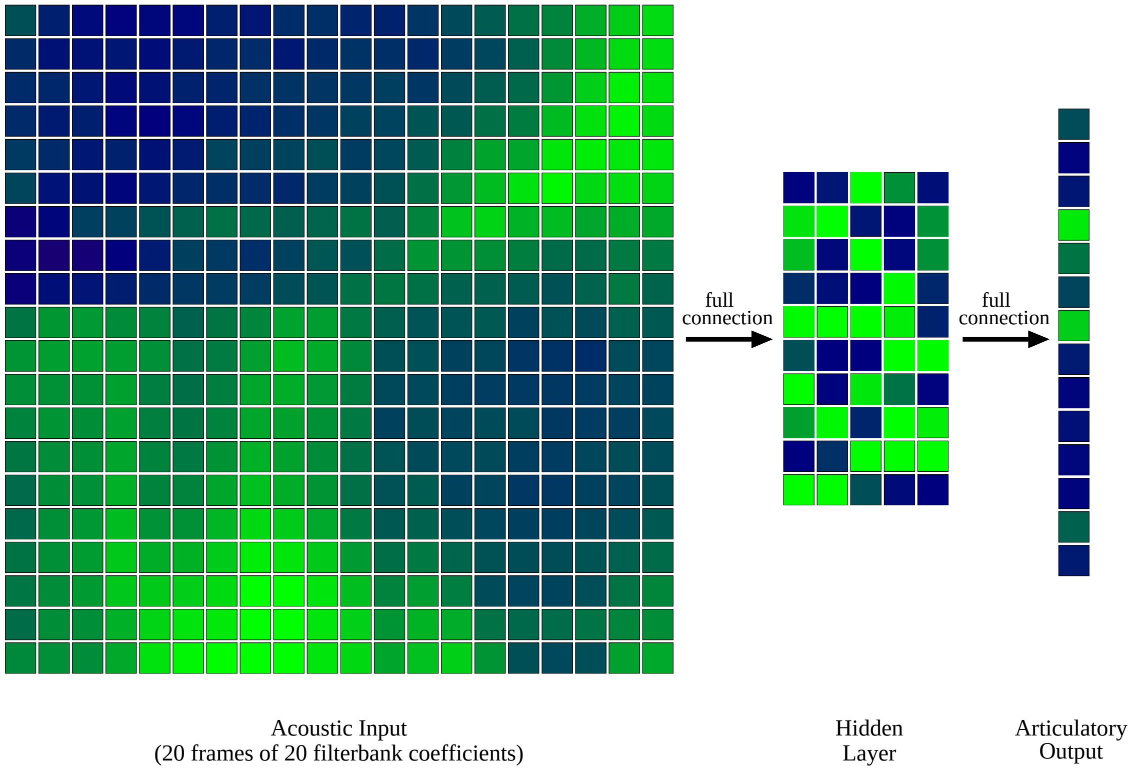

Figure 3.6: This figure shows a feed-forward neural network of the type used to infer articulatory traces from acoustic parameters. A set of 20 filter bank coefficients was computed within 20ms windows on the acoustic signal at 10ms intervals. A context window of 20 such frames provided the input layer of 400 units. The inputs are fully connected to the hidden layer of 50 units, which in turn feeds forward to the 14-dimensional articulatory output. Reproduced from Richmond (2001a) with permission of the author.

Articulatory inversion has not been part of the work of this thesis, and at the time of writing, a set of network-recovered EMA derived using a cross-validation scheme[3] over the complete `fsew0` data is unavailable. Therefore, experiments which incorporate automatically recovered EMA follow the train/test division with which it was created. This is described in Section 5.1.1 on page 116 where the basic classification procedure is introduced.

Combined acoustic-articulatory As well as comparing classification and recognition performance with acoustic and articulatory features, experiments will examine the effect on acoustic-only results of adding articulatory information. Combined feature sets are derived by stacking acoustic MFCC or PLP coefficients and either measured or recovered EMA data.

Numerical differentiation routines from the Edinburgh Speech Tools (Taylor, Caley, Black & King 1997-2003) library were used to calculate the δ and $\delta\delta$ coefficients corresponding to the data for each feature set. Linear discriminant analysis (LDA), which will be described below, was used give reduced dimensionality versions of the larger feature sets.

Linear discriminant analysis

Given a set of p-dimensional vectors, each of which corresponds to 1 of m classes where $p > m$, LDA maps the data into a space of dimensionality at most $m-1$ so as to maximise the separation between classes (Duda & Hart 1973, Balakrishnama & Ganapathiraju 1998). Class-dependent mappings can be found by maximising the ratio of between-class variance to within-class variance, or alternatively, maximising the ratio of overall variance to within-class variance gives a class-independent mapping.

Applying class-dependent mappings would produce speech data which is discontinuous between successive phone segments. Given that part of the motivation of this work is to build a continuous underlying representation of the parameterized speech signal, LDA using class-independent transformations was used to maintain continuity of each feature dimension throughout any given utterance.

[3] K-fold cross-validation is described in the context of phone classification in Section 5.1.1 on page 5.1.1

Letting p_j, $\boldsymbol{\mu}_j$ and Σ_j be the *a priori* probability, mean and covariance of the data which corresponds to class j, the within and between class variance are given by:

$$S_w = \sum_j p_j \, \Sigma_j \tag{3.2}$$

$$S_b = cov(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_j) \tag{3.3}$$

Maximising the variance between classes S_b, whilst minimising that within classes S_w, could be achieved using by transforming the data by matrix $\Omega = S_w^{-1} S_b$. Noting that the construction of Ω entails that it will be of rank $j-1$, dimensionality reduction can be introduced by instead transforming the data by a matrix consisting of the eigenvectors of Ω which correspond to a subset (the larger), or all, of the $j-1$ non-zero eigenvalues.

In this case the classes are the 46 MOCHA phones, and so post-processing with LDA can be applied to give at most a 45-dimensional feature set in the cases where the original dimension is 46 or higher.

Summary of the MOCHA feature sets

Table 3.1 shows a summary of the feature sets which are derived from the MOCHA corpus and will be used for experimentation.

3.2.2 TIMIT

The TIMIT corpus is well known to those working in the field of speech recognition. It was designed and collected specifically for use in development and evaluation of ASR systems and comprises speech data from 630 speakers of eight major dialects of American English. Each subject speaks 10 phonetically rich sentences, of which 2 are the same for all speakers and are included to highlight dialectical variation. These are ignored for all experimental work as inclusion would skew the phone coverage. As for the MOCHA corpus, MFCC and PLP cepstral coefficients, energy and corresponding δ and δ parameters were generated from the waveform using HTK. Table 3.1 gives a summary of the feature sets which are derived from the TIMIT corpus for use in later experimentation.

type	feature	plain	$+\delta$	$+\delta + \delta\delta$	$+\delta$ LDA	$+\delta + \delta\delta$ LDA
acoustic	MFCC	√	√	√		
	PLP	√	√	√		
articulatory	EMA	√	√	√		
	EMA + EPG + LAR	√	√	√		√
	net EMA	√	√	√		
combined	MFCC + EMA	√	√		√	√
	MFCC + net EMA	√	√		√	√
	PLP + EMA	√	√		√	√
	PLP + net EMA	√	√		√	√

Table 3.1: Each tick denotes a feature set which will be used for MOCHA speaker-dependent classification. All experiments use the data from speaker `fsew0`, and results are presented in Section 5.1. Log signal energy (and corresponding derivatives where required) are appended to both of the acoustic features.

type	feature	plain	$+\delta$	$+\delta + \delta\delta$
acoustic	MFCC	√	√	√
	PLP	√	√	√

Table 3.2: Each tick denotes a feature set which will be used for TIMIT speaker-independent classification and recognition. Log signal energy (and corresponding derivatives where required) are appended to both features.

3.3 Language Modelling

The summary of the component parts of a speech recognizer on page 2 stated that a language model was incorporated to give an estimate of the prior probability of each candidate word sequence, $w_1^j = \{w_1, \ldots, w_j\}$. Assuming that the probability of any given word depends only on the identities of the words which precede it, Bayes' rule can be used to decompose $P(w_1^j)$ as:

$$P(w_1^j) = \prod_{i=1}^{j} P(w_i | w_1 \ldots w_{j-1}) \tag{3.4}$$

An n-gram language model assumes that the probability of any given word is only dependent on the preceding $n-1$ words, so the relation 3.4 is reduced to:

$$P(w_1^j) = \prod_{i=1}^{j} P(w_i|w_{i-1}, \ldots, w_{i-n+1}) \tag{3.5}$$

Setting $n = 3$ gives a trigram, which has proven to be an extremely effective means of language modelling for ASR. Bigrams, for which $n = 2$, are commonly used for tasks in which the acoustic modelling is the primary concern. For example, phone recognition on the TIMIT corpus is a benchmark test for which it is usual to report results for systems including bigram language models.

Frequency counts of occurrences in the training data can be used to estimate n-gram probabilities. However, even for small vocabularies there are commonly n-grams which either do not appear in the training set or do not occur enough times to give robust estimates of their probabilities. Discounting and backing-off are common methods of smoothing in such situations. Discounting involves shifting probability mass from common to infrequently seen n-grams, and backing off provides a method of replacing missing n-grams with rescaled probabilities of lower order word sequences. For instance, if the language model was of trigrams, and the sequence $\{w_{i-2}, w_{i-1}, w_i\}$ not adequately estimable, backing off would set:

$$\hat{P}(w_i|w_{i-1}, w_{i-2}) = P(w_i|w_{i-1})B(w_{i-1}, w_{i-2}) \tag{3.6}$$

where $B(w_{i-1}, w_{i-2})$ weights $P(w_i|w_{i-1})$ so as to normalised the probability mass of the trigrams finishing with w_i (Young 1995).

The language models used in the phone classification experiments of Chapter 5 are simple bigrams with no backing off. Code from the Edinburgh Speech Tools (Taylor et al. 1997-2003) was used to estimate probabilities by counting occurrences of phone pairs in the training data. A minimum probability is set to avoid zeros for phone pairs which do not occur in the training set. The phone recognition experiments which follow in Chapter 6 use backed off phone bigrams, with probabilities again estimated on the training data. These language models were produced using the CMU-Cambridge Statistical Language Modelling toolkit (Clarkson & Rosenfeld 1997).

3.4 Linear Predictability

This chapter has so far introduced the data and some of the basic techniques which will be required for experiments presented in later chapters. The remainder of the chapter involves some preliminary data analysis. One of the attributes of the ideal acoustic model for ASR as given on page 3 was to account for the temporal dependencies present in speech data. The following sections examine these dependencies and provide empirical motivation for applying the model at the core of this work to speech data.

Linear models are in general simpler to deal with than their non-linear counterparts. There are comparatively few functional forms to choose between and the properties of linear models are straightforward and well known. A commonly accepted approach to model selection is to choose the simplest model which can describe the process in question. With this in mind, the experiments below examine the suitability of linear models for accounting for the dependencies present in the speech data used in this thesis, acoustic and articulatory. Modelling of the correlations occurring within and between phonetic segments are considered separately.

Digalakis (1992) carried out such an experiment, examining whether linear models can account for the relationships present in an MFCC parameterization of speech. The conclusion was that linear models are appropriate *intra*, but not *inter* phone. Results of similar experiments are presented below in greater detail, and extended to assess linear models of articulatory traces and of a PLP cepstra parameterization of the acoustic signal. Each of the core data/feature sets used in this thesis is examined to determine whether or not assumptions of linearity are appropriate. Results are further broken down by phonetic class to enable comparison with classification and recognition accuracies presented later in the thesis. Appendix A.1 gives these classes along with the IPA symbols corresponding to the MOCHA and TIMIT phone sets.

3.4.1 Method

The approach will be to compare the fit of linear and non-linear regressions on subsets of speech data. If a linear regression can account for as much of the systematic variation as a non-linear counterpart, there is no justification for the extra complexity of employing a non-linear model.

Letting $\mathbf{y}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$ be random variables, with $\mathbf{y}$ the dependent variable and $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$ as predictors, the linear regressions use a standard multiple regression model, such that

$$\mathbf{y} = \sum_{i=1}^{p} \alpha_i \mathbf{x}^{(i)} + \beta. \tag{3.7}$$

The non-linear regressions use a generalized additive model (Hastie & Tibshirani 1986):

$$\Theta(\mathbf{y}) = \sum_{i=1}^{p} \Phi_i(\mathbf{x}^{(i)}) \tag{3.8}$$

Here, $\Theta, \Phi_1, \ldots, \Phi_p$ can take a variety of forms including non-linear transformations. This model is not a fully general non-linear mapping (such as given by a neural network), though is chosen as non-linear dependencies can be accounted for, and the two models are of analogous form. It is apparent that the linear regression model of Equation 3.7 is a special case of the generalized additive model. The alternating conditional expectation (ACE) algorithm (Breiman & Friedman 1985) is used to choose an optimal set of transformations, $\Theta^*, \Phi_1^*, \ldots, \Phi_p^*$ which minimise the fraction of variance not explained by the regression,

$$e^2(\Theta, \Phi_1, \ldots, \Phi_p) = \frac{E\left[\left(\Theta(\mathbf{y}) - \sum_{i=1}^{p} \Phi_i(\mathbf{x}^{(i)})\right)^2\right]}{E[\Theta^2(\mathbf{y})]}. \tag{3.9}$$

The two regression models are compared using the R^2 statistic, which gives the fraction of variability in $\mathbf{y}$ that is accounted for by the regression on $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$. It is computed using the residual and data sums of squares:

$$R^2 = 1 - \frac{SS_{residual}}{SS_{data}} \tag{3.10}$$

Within, or *intra*, segmental dependencies are examined on a phone by phone basis. Time-aligned phonetic labels are first used to extract all the instances of each phone type. Both linear and non-linear regressions are carried out with the central frame of a segment as the dependent variable. Separate regressions are carried out with the explanatory variable as either the data from the frame preceding the central one, or the segment-initial one. In each case, an R^2 value averaged across the dimensions of the feature vector is produced for each phone. These are combined in proportion with relative phone frequencies to give an overall value. As well as looking at absolute differences in R^2

between the two types of regression, a measure of relative performance is calculated. $R^2_{linear}/R^2_{non-linear}$ is computed for each feature dimension of each phone, combined as above in proportion to phone frequencies, and multiplied by 100 to give an overall percent relative performance.

Producing MFCCs and PLP cepstra was discussed above in Section 3.1.2. Since analysis is made at 10ms intervals within 25ms windows on the acoustic signal, there will be an overlap in the information conveyed by successive frames. Similarly for the EMA data, downsampling from the original frame-shift of 2ms to give the frame-shift of 10ms used in the experiments below includes a low-pass filtering step which will introduce some smearing of information between consecutive frames. Therefore, it is expected that regressions of the segment-central frame on the preceding frame will give much higher absolute R^2 values regardless of model type than for those regressions which use the segment-initial frame as the dependent variable. However, it is *relative* performance of the linear and non-linear models on which the suitability or otherwise of linear models will be decided.

MOCHA			TIMIT		
phone pair	frequency	example	phone pair	frequency	example
[`dh-@`]	260	**th**e	[`ix-n`]	2208	th**in**
[`s-t`]	160	**st**op	[`ao-r`]	1019	**or**
[`y-uu`]	150	**you**	[`l-iy`]	928	quick**ly**
[`i-n`]	132	b**in**	[`ix-z`]	854	f**izz**
[`n-d`]	120	a**nd**	[`dh-ax`]	852	**the**
[`r-i`]	101	**ri**nse	[`t-r`]	586	**tr**ee

Table 3.3: Commonly occurring phone pairs which were used to examine inter-phone dependencies present in MOCHA and TIMIT. Appendix A.1 gives the IPA symbols for each of the phone classes used to label the MOCHA and TIMIT corpora.

The 6 commonly occurring phone pairs given in Table 3.3 were chosen for use in experiments to compare linear and non-linear models between, or *inter* segments. For each of the phone pairs, the instances in which the second phone in the pair is 5 frames or longer are used in experimentation, and *intra*-phone regressions of the 5^{th} on the 2^{nd}

frame of the second phone are performed. Then, the 2^{nd} frame of the second phone is taken as the dependent variable, with the penultimate frame of the first phone in the pair used as the explanatory variable for an *inter*-phone regression. These form pairs of inter-phone and intra-phone regressions in which the interval between the dependent and explanatory variables is kept constant. In this way, any confounding factors due to the spacing between regression variables are controlled for, allowing a direct examination of the effect of crossing phone boundaries. Results are compiled in a similar fashion to the within-phone comparisons, replacing relative phone frequencies with relative phone pair frequencies.

3.4.2 Results

Results take the form of R^2 values for each feature dimension of each phone (or phone-pair), for both linear and non-linear regressions. This correspondence makes a paired t-test as described in Section 5 of Chapter 6 a natural choice to check the hypothesis of equality of fit under each of the two models. For every experiment in this section, this was refuted in no uncertain terms. In other words, the non-linear regression was always a better fit to the data than the linear one. However, whilst linear models do not perform exactly as well as their non-linear counterparts, in many cases they are extremely close.

MOCHA results

The within-segment experiments show that under both models, R^2 values are considerably higher for regression on the previous frame than on the initial frame, much as expected. On the articulatory data, the overall linear and non-linear R^2 values are 0.981 and 0.987 respectively for the regressions on the previous frame. Not only are these both extremely high, they are very close. When the regression uses the segment-initial frame as the explanatory variable, the linear model is outperformed by a larger margin, though still gives 91.1% of the non-linear R^2. Acoustic features do not provide such consistently well fitting models, with R^2 values ranging from 0.815 for non-linear regression on the preceding frame for MFCCs, down to 0.301 for linear regression on the segment-initial frame for PLPs. MFCC features provide a better fit, and a higher relative linear versus

features	explanatory variable	R^2 linear	R^2 non-linear	linear performance as percent of non-linear
EMA	previous	0.981	0.987	99.4%
	initial	0.797	0.875	91.1%
MFCC	previous	0.740	0.815	90.7%
	initial	0.343	0.550	62.5%
PLP	previous	0.678	0.780	86.9%
	initial	0.301	0.535	56.2%

Table 3.4: Results of regressions to compare the performance of linear and non-linear regressions in predicting dependencies within phones. R^2 values averaged over 46 phone classes for linear and non-linear regressions which predict segment-central frames based either on previous or segment-initial frames. Also shown are percentages of non-linear R^2 gained by linear regressions. Data is from the speaker `fsew0` in the MOCHA corpus

non-linear performance, than the PLP features do. In both cases the linear models manage to capture a high proportion of the dependencies accounted for by non-linear regressions when the previous frame is used as the dependent variable, 90.7% and 86.9% for MFCCs and PLPs respectively. However, relative performance is substantially worse when the initial frames are used as predictors.

These results are shown pictorially in Figure 3.7, broken down by phonetic class. The approximately parallel graphs of R^2 values for the two acoustic features show that the internal structure of which phone types are modelled well are similar for both. Overall though, regression models can produce better fits to MFCCs than PLPs. For all feature types, regressions on the previous frame within diphthongs produce some of the largest R^2 values and highest relative performances of linear models against their non-linear counterparts. However, these become the lowest relative and some of the lowest absolute scores when the initial frame becomes the explanatory variable. This is to be expected as diphthongs are by their nature transitional and so subject to a high degree of variation. In general, it is for vowels, nasal stops, and liquids that linear models give values of R^2 most similar to their non-linear counterparts, and affricates where the differences are largest.

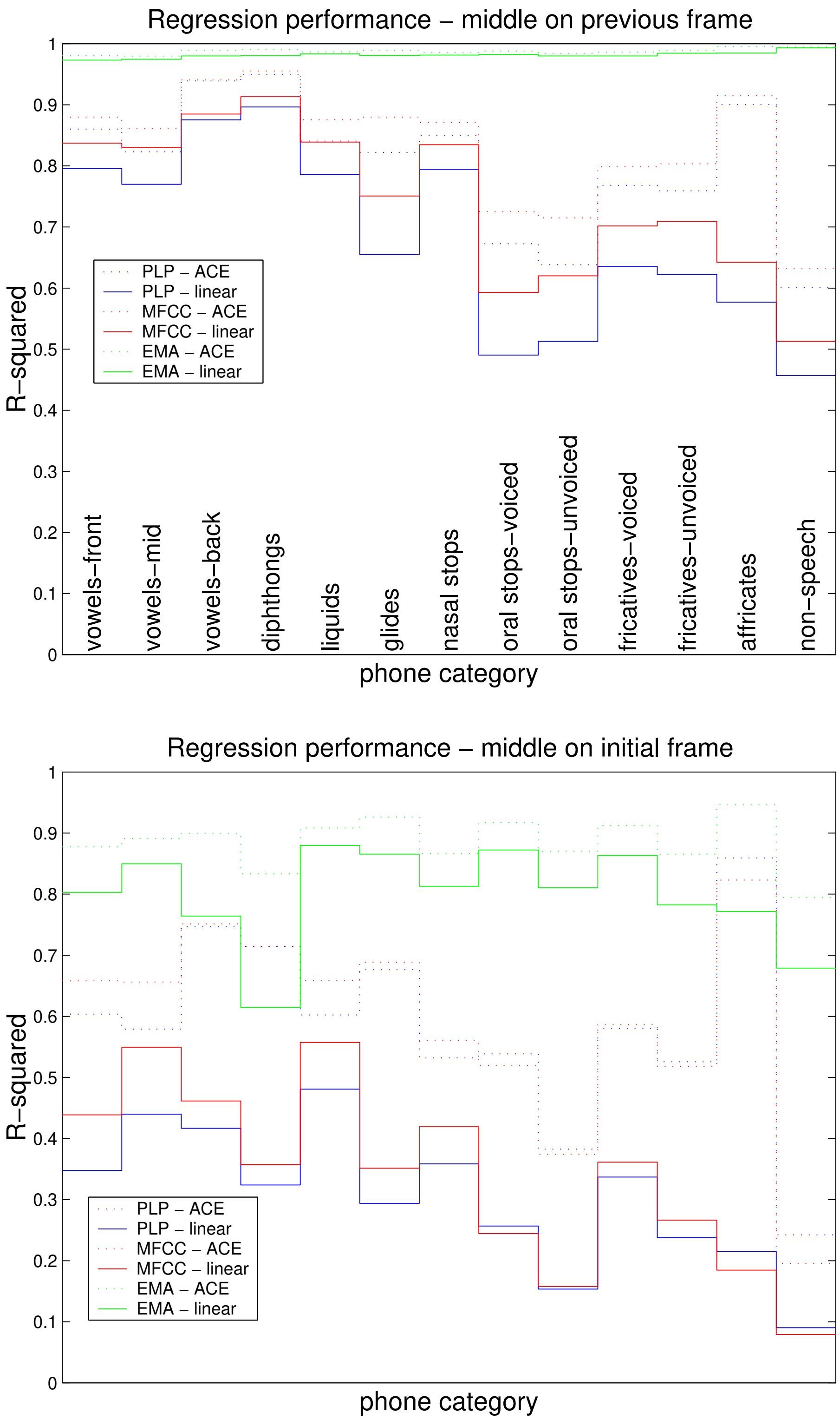

Figure 3.7: Pictorial comparison of the proportion of variance accounted for by linear and non-linear regression, where results for individual phones have been pooled into phonetic categories as given in Appendix A.1. Data comprises EMA articulatory traces and two parameterizations of the acoustic signal, MFCCs and PLPs, for a single speaker from the MOCHA corpus. The phone categories in the lower graph match those in the upper.

features	regression	R^2 linear	R^2 non-linear	linear performance as percent of non-linear
EMA	intra	0.917	0.990	92.6%
EMA	inter	0.902	0.989	91.2%
MFCC	intra	0.519	0.963	53.9%
MFCC	inter	0.462	0.958	47.9%
PLP	intra	0.548	0.973	56.3%
PLP	inter	0.489	0.964	50.5%

Table 3.5: Results of regressions to examine the effect of crossing phone boundaries on linear and non-linear regressions. R^2 values averaged over 6 frequently occurring phone pairs are given for linear and non-linear regressions on intra-phone and inter-phone bases. Also shown are percentages of non-linear R^2 gained on linear regressions. Data is from the speaker `fsew0` in the MOCHA corpus, and the phone pairs used are given in table 3.3.

Table 3.5 shows the results comparing the pairs of inter-phone and intra-phone regressions. In all cases, linear models give performance closer to that of the non-linear models when regressions are on an intra-phone rather than inter-phone basis. This effect is smallest for articulatory data for which the linear models give 92.6% and 91.2% of the R^2 value found under non-linear models for intra-phone and inter-phone regressions respectively. On acoustic data, the R^2 values found using linear models are considerably lower than those found with their non-linear counterparts, with linear models giving in the region of 50% of the performance of non-linear models. The relative R^2 values given by linear and non-linear models are 53.9% and 47.9% on intra-phone and inter-phone regressions respectively with MFCCs as features. This decrease is largely due to the linear regression R^2 value reducing from 0.519 to 0.462 where its non-linear equivalent only drops by 0.005 from 0.963 to 0.958. These, and similar results for PLPs, suggest that the effects of crossing phone boundaries are more detrimental to the performance of linear than non-linear predictors.

These results also support those given in Table 3.4 in demonstrating the poor relative performance of linear compared to non-linear models when the regression variables are spaced a number of frames apart.

TIMIT results

features	explanatory variable	R^2 linear	R^2 non-linear	linear performance as percent of non-linear
MFCC	previous	0.690	0.717	96.2%
	initial	0.398	0.442	90.0%
PLP	previous	0.706	0.734	96.1%
	initial	0.407	0.456	89.2%

Table 3.6: Results of regressions to compare the performance of linear and non-linear regressions in predicting dependencies within phones. R^2 values averaged over 61 phone classes for linear and non-linear regressions which predict segment-central frames based either on preceding or segment-initial frames. Also shown are percentages of non-linear R^2 gained by linear regressions. Data comprises the training set from the speaker-independent TIMIT corpus

The TIMIT data results show many of the same trends as seen in the MOCHA data, though estimations should be more reliable as there is considerably more data available. Model fits using PLP and MFCC features are again close for the between-segment experiments, though in a role reversal, the PLP data provides better regression models than the MFCCs. Linear regression of central on preceding frame gives R^2 values of 0.706 and 0.690 for the PLP and MFCC features respectively, compared to non-linear values of 0.734 and 0.717. These both represent over 96% of the non-linear fit.

As before, R^2 values are significantly lower when the segment-initial rather than preceding frame is used as predictor. Non-linear regressions give R^2 values of 0.456 and 0.442 for PLPs and MFCCs respectively, and linear models manage about 90% of these. The graph in Figure 3.8 shows these results broken down by phonetic category. PLP and MFCC features exhibit similar trends, with the best fits and smallest differences between linear and non-linear model fit for vowels, liquids and nasal stops with the preceding frame as the predictor. Just as with the MOCHA data, diphthongs give high relative performance with the previous frame as predictor, and considerably lower relative performance using the segment-initial frame.

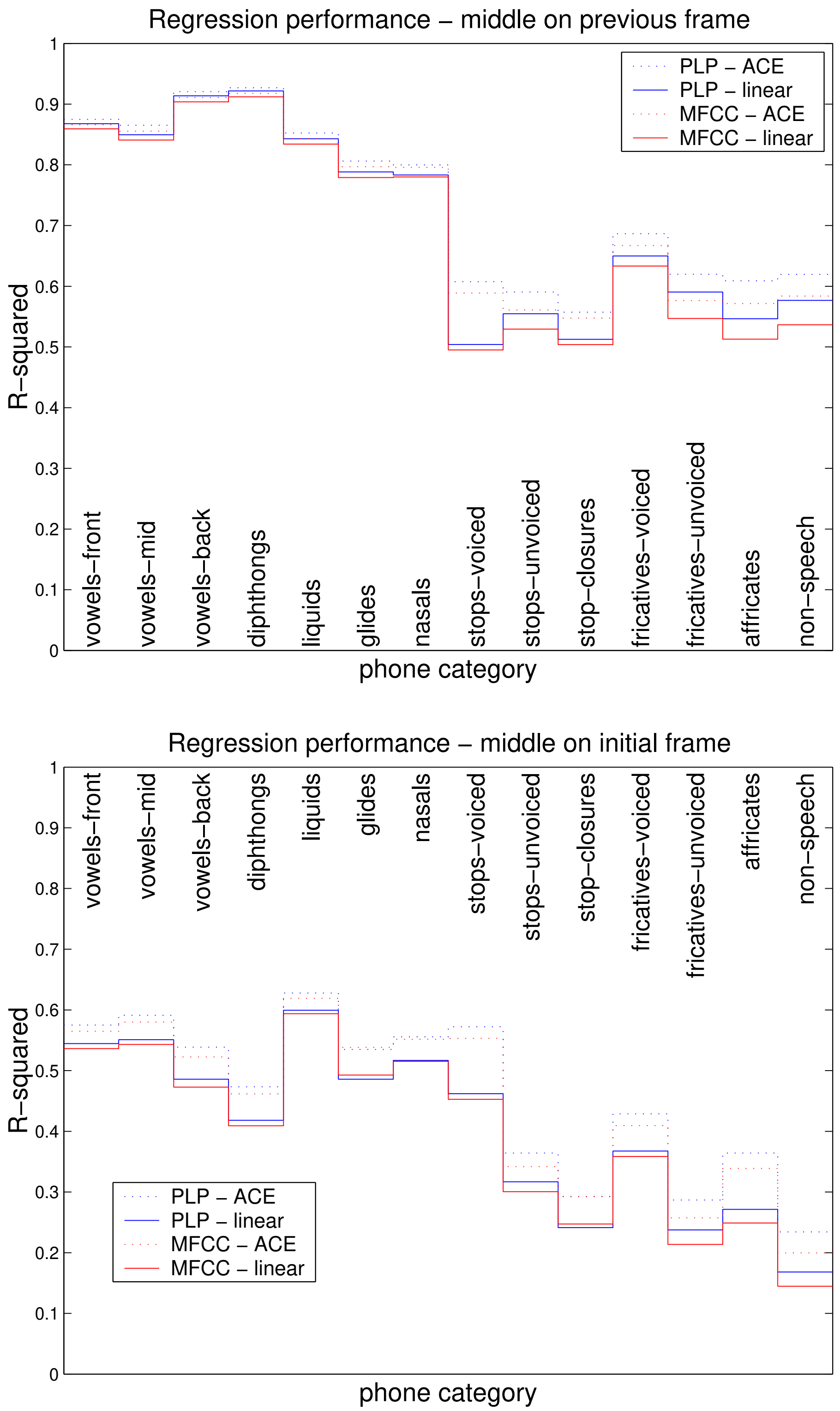

Figure 3.8: Pictorial comparison of the proportion of variance accounted for by linear and non-linear regression, where results for individual phones have been pooled into phonetic categories as given in Appendix A.1. Data comprises two parameterizations of the acoustic signals from the training set of the multi-speaker TIMIT corpus, MFCCs and PLP cepstra.

features	regression	R^2 linear	R^2 non-linear	linear performance as percent of non-linear
MFCC	intra	0.543	0.645	84.2%
	inter	0.406	0.561	72.1%
PLP	intra	0.553	0.661	83.7%
	inter	0.415	0.573	72.4%

Table 3.7: Results of regressions to examine the effect of crossing phone boundaries on linear and non-linear regressions. R^2 values averaged over 6 frequently occurring phone pairs are given for linear and non-linear regressions on intra-phone and inter-phone bases. Also shown are percentages of non-linear R^2 gained on linear regressions. Data is from the training set of speaker-independent TIMIT corpus, and the phone pairs used are given in Table 3.3.

Table 3.7 shows the results of the regressions used to examine the effect of crossing phone boundaries on linear and non-linear regression models. For both MFCCs and PLPs, absolute R^2 values are higher for intra-phone than for inter-phone regressions regardless of the regression model. Also, the proportion of the non-linear R^2 given by the linear model is higher when regressions are on an intra-phone basis. For example, with MFCC features, the relative intra-phone performance of linear regression models is 84.2% which drops to 72.1% when the regressions are inter-phone. As with the MOCHA data, these results suggest that crossing phone boundaries is more detrimental to the performance of a linear predictor than a non-linear counterpart.

Conclusions

All the speech data extracted from the MOCHA corpus came from a single speaker. It was expected that the consistency across such a data-set would lead to high R^2 values compared with those on TIMIT in which there are a multitude of different vocal characteristics and speaking styles. This was true for the non-linear regressions on acoustic features, however with the exception of using the preceding frame to predict the segment-central frame for MFCC features, linear models consistently produced lower absolute intra-phone R^2 values on MOCHA data than on TIMIT. Furthermore, the relative performance of linear compared to non-linear regressions was lower for MOCHA acoustic data than for

TIMIT equivalents.

Comparing the pairs of inter-phone and intra-phone results in Tables 3.5 and 3.7 shows that non-linear models account for more of the variation in the dependent variable than linear models, when the regression variables are spaced at an interval of 3 frames. The results in these tables further demonstrate that the relative performance of linear compared to non-linear models is reduced when the regressions cross phone boundaries. This suggests that linear predictors are not suited to modelling inter-phone dependencies.

On MOCHA speaker-dependent data, linear and non-linear models give comparable fits predicting central using preceding frame for all features, and a comparable fit predicting central using initial frame with articulatory data. Given the extremely good fit of the linear model to the articulatory data, a linear model seems entirely suitable in this case. Furthermore, given that 98% of the variation in the phone-central frame could be explained using the preceding one, a first order model such as the regression model in Equation 3.7 seems ideal. This finding supports that of Roweis (1999) who shows that linear models with only a few degrees of freedom can account for much of the structure present in articulatory data. Linear models also seem adequate for the acoustic data when the preceding frame is used as predictor. However, the more general conclusions on model choice for acoustic data based on these results will be based on the speaker-independent TIMIT experiments.

Intra-phone regressions on TIMIT acoustic data show that with the preceding frame as the explanatory variable, application of a linear model gives 96% of the fit of a non-linear equivalent. Furthermore, a linear regressor accounts for over 70% of the variation in the data. The success of such a simple regression model, and the closeness to the fit of a non-linear model, justifies the exploration of a first-order linear model of the parameterized speech signal on an intra-phone basis.

Chapter 4

Linear Dynamic Models

This chapter describes the linear dynamic model in detail: the function of each component of the model, parameter estimation, evaluation, the assumptions made in applying LDMs to speech data, and the variations which will be compared experimentally in Chapter 5.

4.1 The LDM and its component parts

The class of state-space models, to which the LDM belongs, was introduced in Section 2.1.1 on page 13. However, it is worth re-stating the purpose of such a model, which is to make a distinction between the underlying process and the observations with which it is represented. With $\mathbf{y}_t$ and $\mathbf{x}_t$ representing p and q dimensioned observation and state vectors respectively, an LDM is specified by the following pair of equations:

$$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{4.1}$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t \qquad \boldsymbol{\eta}_t \sim N(\mathbf{w}, D) \tag{4.2}$$

and a distribution over the initial state, $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$. The LDM assumes that the dynamics underlying the data can be accounted for by the autoregressive *state process* 4.2. This describes how the Gaussian-shaped cloud of probability density representing the state evolves from one time frame to the next. A linear transformation via the matrix F and the addition of some Gaussian noise, $\boldsymbol{\eta}_t$, provide this, the dynamic portion of the model. The complexity of the motion that Equation 4.2 can model is determined by the dimensionality of the state variable, and will be considered below. The *observation* *process* 4.1 shows how a linear transformation with the matrix H and the addition of measurement noise $\boldsymbol{\epsilon}_t$ relate the state and output distributions.

4.1.1 State process

Practical use of an LDM involves filtering or smoothing to provide estimates of the state vectors dependent on a set of observed values. However, to build a clear picture of the model's capabilities, we first consider the state process in isolation.

The transform F consists of a combination of rotations and stretches about and along the dimensions of the state-space, and the noise element is additive, given by the Gaussian $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$. The mean $\mathbf{w}$ can be non-zero, allowing for a steady drift of the state vector. With $\mathbf{x}_t$ and Σ_t representing the mean and covariance of the state distribution at time t, applying the update equation can be seen as consisting of two elements. The first is a linear transformation, $\mathbf{x}_t = F\mathbf{x}_{t-1}$, in which the Gaussian distribution of $\mathbf{x}_t$ is preserved but rescaled giving:

$$\mathbf{x}_t \sim N(F\mathbf{x}_{t-1},\ F\Sigma_t F^T) \tag{4.3}$$

and the second is convolution with the state error $\boldsymbol{\eta}_t$. The result of convolving a pair of Gaussian random variables $z_1 \sim N(\mu_1, \theta_1)$ and $z_2 \sim N(\mu_2, \theta_2)$ is also Gaussian:

$$z_1 + z_2 \sim N(\mu_1 + \mu_2,\ \theta_1 + \theta_2) \tag{4.4}$$

and so the evolution of the state distribution is therefore:

$$\mathbf{x}_t \sim N(F\mathbf{x}_{t-1} + \mathbf{w},\ F\Sigma_t F^T + D). \tag{4.5}$$

The state dimension determines the nature of the dynamics which the system can model. This comes about as the potential for interaction between dimensions increases with the size of the state vector. With a state dimension of 1, the model can describe exponential growth or decay with some general trend. Figure 4.1 shows plots of the state means for two such models, produced by generating values according to the state Equation 4.2. The parameters used were:

$$F = [0.95]\,, \mathbf{w} = [0.008]\,, \boldsymbol{\pi} = [0.5] \ \text{ and } F = [0.88]\,, \mathbf{w} = [0.11]\,, \boldsymbol{\pi} = [0.5]$$

in the first and second plots respectively. To show the variance changing over time, single standard deviations from the mean are also included in the figures. In both cases, the variances were set with $D = 0.005$ and $\Lambda = 0.05$.

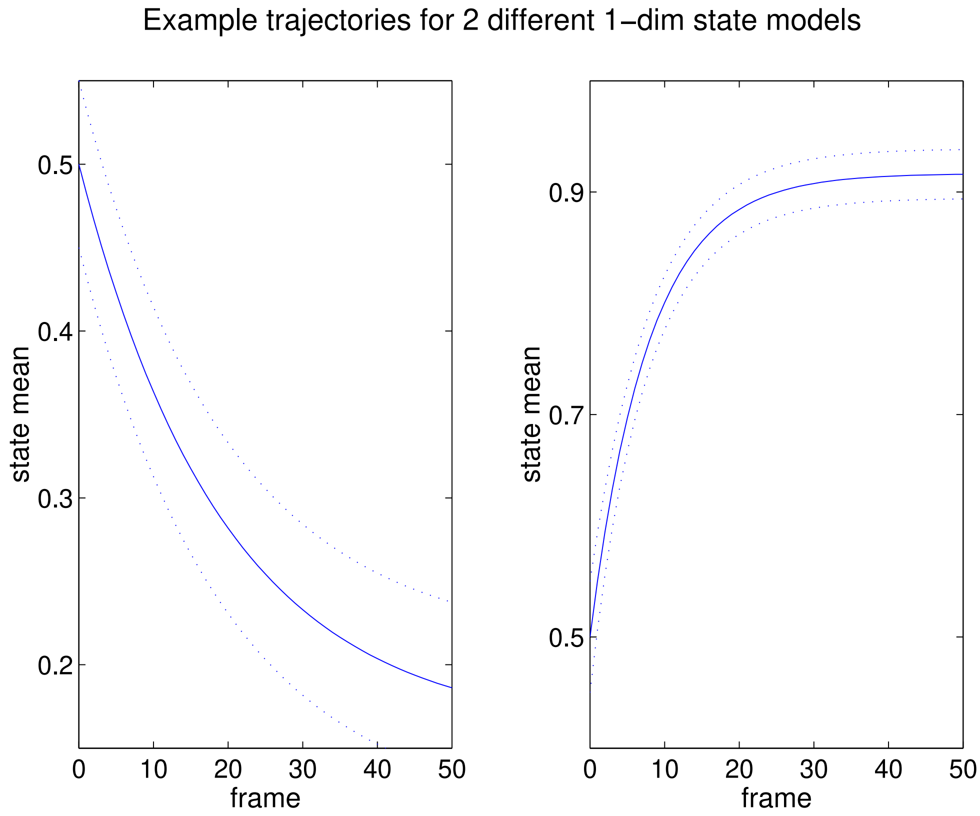

Figure 4.1: A 1-dimension state can model exponential growth or decay, with a steady drift provided by the mean of the state error. This plot shows state mean against time for two examples of 1-dimensional models. The dotted lines are placed a single standard deviation from the mean and show how the variance evolves as the model runs.

With a state dimension of 2, the transform F can be composed of rotations and rescaling along the coordinates of 2-space. In this case, state trajectories describe damped or exponentially increasing oscillations, again with an overall drift. Figure 4.2 shows the two state axes over time for such a model. State means were again generated according to Equation 4.2, this time with the parameters:

$$F = \begin{bmatrix} 0.9 & 0.2 \\ -0.1 & 0.99 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 0.02 \\ 0.005 \end{bmatrix}, \boldsymbol{\pi} = \begin{bmatrix} 0.12 \\ 0.10 \end{bmatrix}$$
$$D = \begin{bmatrix} 0.0005 & 0.0 \\ 0.0002 & 0.00012 \end{bmatrix}, \Lambda = \begin{bmatrix} 0.0005 & -0.0004 \\ -0.0004 & 0.0008 \end{bmatrix} \tag{4.7}$$

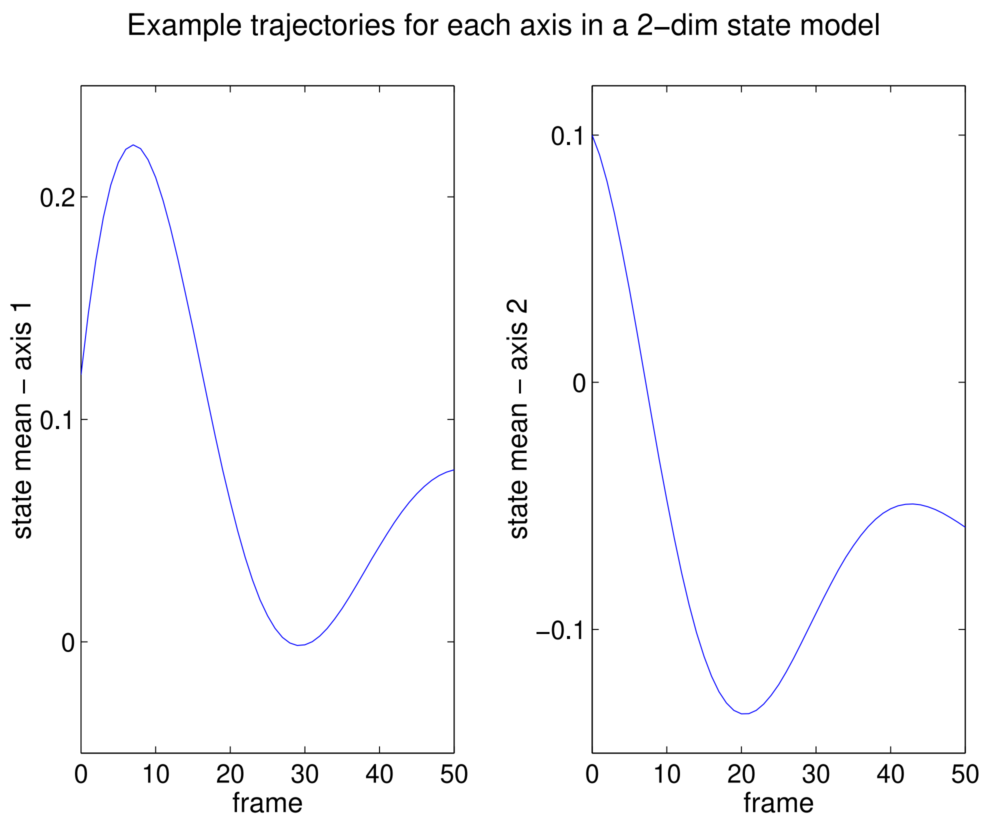

Figure 4.2: A 2-dimensional state can produce damped oscillations, again with an overall drift which is provided by the state error mean. This figure shows each axis of the state vector plotted against time for an example of a 2-dimensional model. The second axis plot resembles a phase-shifted rescaling of the first.

A 2-dimensional state provides a good opportunity to visualise the evolution of the cloud of probability density surrounding the mean. Figure 4.3 shows the same state process as in Figure 4.2, this time with one dimension plotted against the other and ellipses showing a single standard deviation around the mean at $t = 0$, $t = 15$, $t = 30$, and $t = 45$. The figure illustrates how the principal axes of the covariance rotate so that the probability density can preserve its shape around the direction of flow.

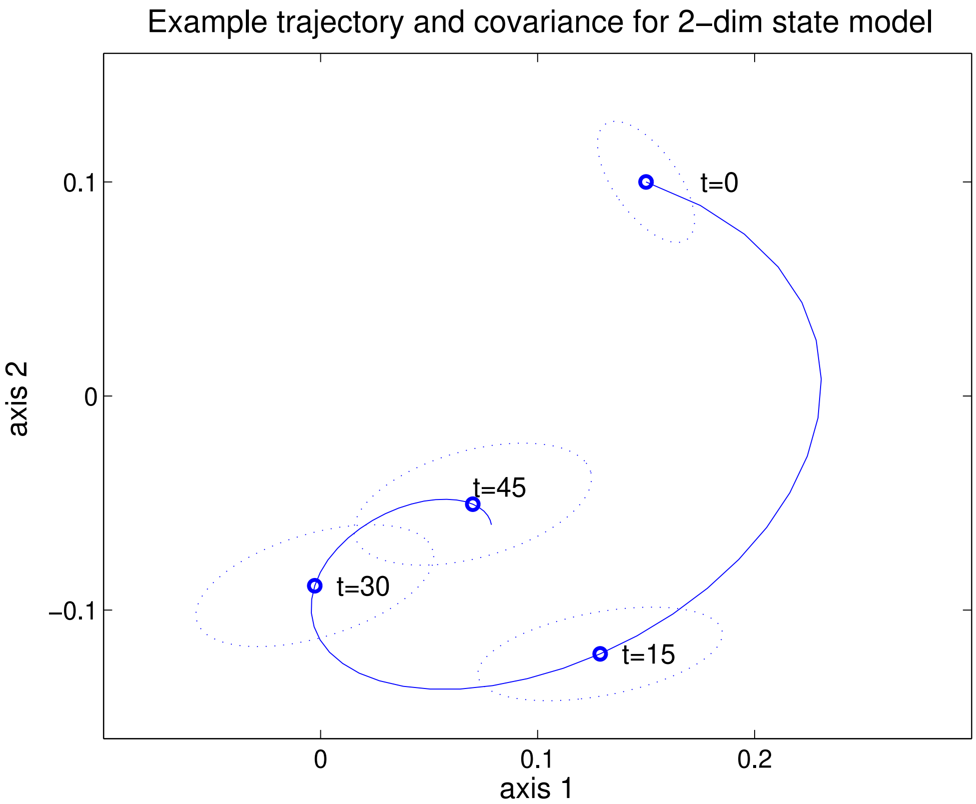

Figure 4.3: The state dimensions shown in Figure 4.2 are plotted here against one another. The ellipses show a single standard deviation around the mean at $t = 0$, $t = 15$, $t = 30$, and $t = 45$. This figure illustrates how the principal axes of the distribution rotate so that the density can preserve its shape around the direction of flow.

With larger state dimensions, the interactions between axes provide the capacity to model more complex oscillations. Figure 4.4 shows each of the dimensions in an example of a 4-dimensional model over time, generated using

$$F = \begin{pmatrix} 0.90 & -0.04 & -0.01 & -0.06 \\ 0.02 & 0.92 & 0.03 & 0.13 \\ -0.03 & -0.02 & 0.83 & -0.01 \\ 0.09 & -0.21 & -0.01 & 0.92 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 0.01 \\ 0.1 \\ 0.21 \\ -0.1 \end{pmatrix} \boldsymbol{\pi} = \begin{pmatrix} 0.2 \\ 0.0 \\ 1.1 \\ 0.2 \end{pmatrix}$$

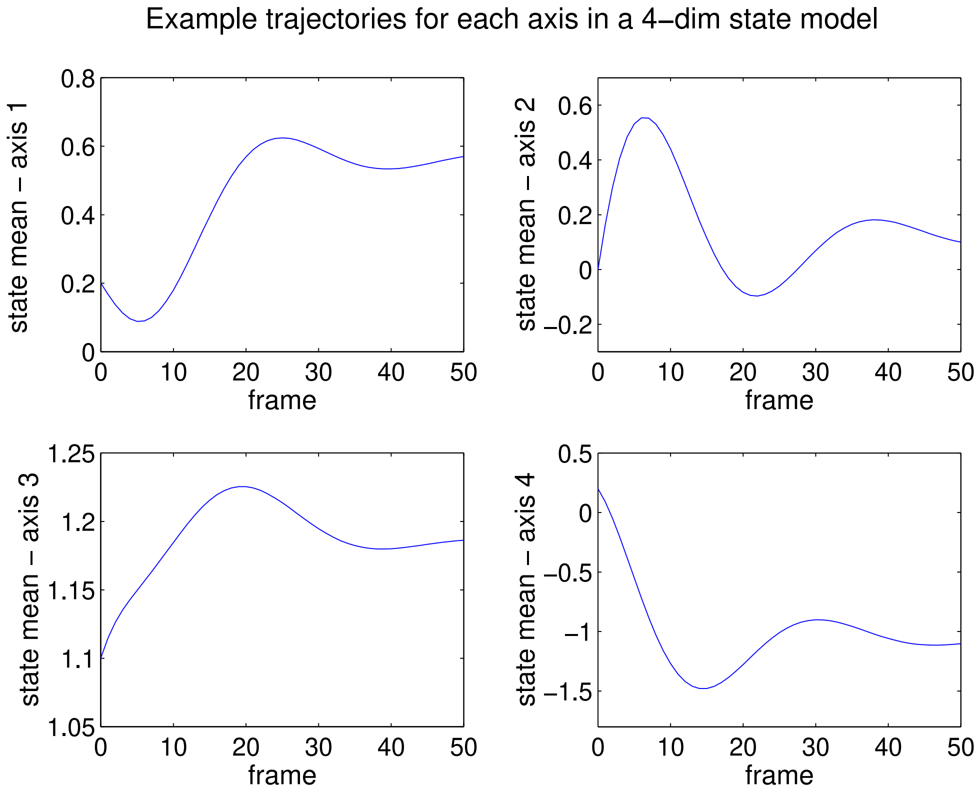

Figure 4.4: A higher dimension state-space can contain more complexity within each period of oscillation, and hence model more complicated trajectories. Here the axes of the state mean for an example of a 4-dimensional model are plotted over time.

Target nature of the state evolution

In this work, the evolution matrix F will be constrained to be a decaying mapping (see Section 4.2.4 on page 99), and so the state trajectories are destined to converge. Given

$$\hat{\mathbf{x}}_t = F\hat{\mathbf{x}}_{t-1} + \mathbf{w} \tag{4.9}$$

and for $|F| < 1$,

$$\exists\, \hat{\mathbf{x}}_\infty \text{ s.t. } \lim_{t\to\infty} \hat{\mathbf{x}}_t = \mathbf{x}_\infty \tag{4.10}$$

$\hat{\mathbf{x}}_\infty$ can then be solved for using Equation 4.9:

$$\hat{\mathbf{x}}_\infty = F\hat{\mathbf{x}}_\infty + \mathbf{w} \tag{4.11}$$

$$\Rightarrow \hat{\mathbf{x}}_\infty = (I - F)^{-1}\mathbf{w} \tag{4.12}$$

This gives an interesting insight into the workings of the dynamic portion of the LDM. Since the constraint is made that $|F| < 1$, the state's evolution is governed by a set of accelerations and velocities with which to attain some steady-state location in state-space.

Figure 4.5 shows the state trajectories of Figure 4.4, with a dashed line along the target mean, as found in 4.12, for each. For each of the four state dimensions, the state is shown tending toward its predicted target. Another visualisation of the same model is given in Figure 4.6, where pairs of the state dimensions are plotted against each other.

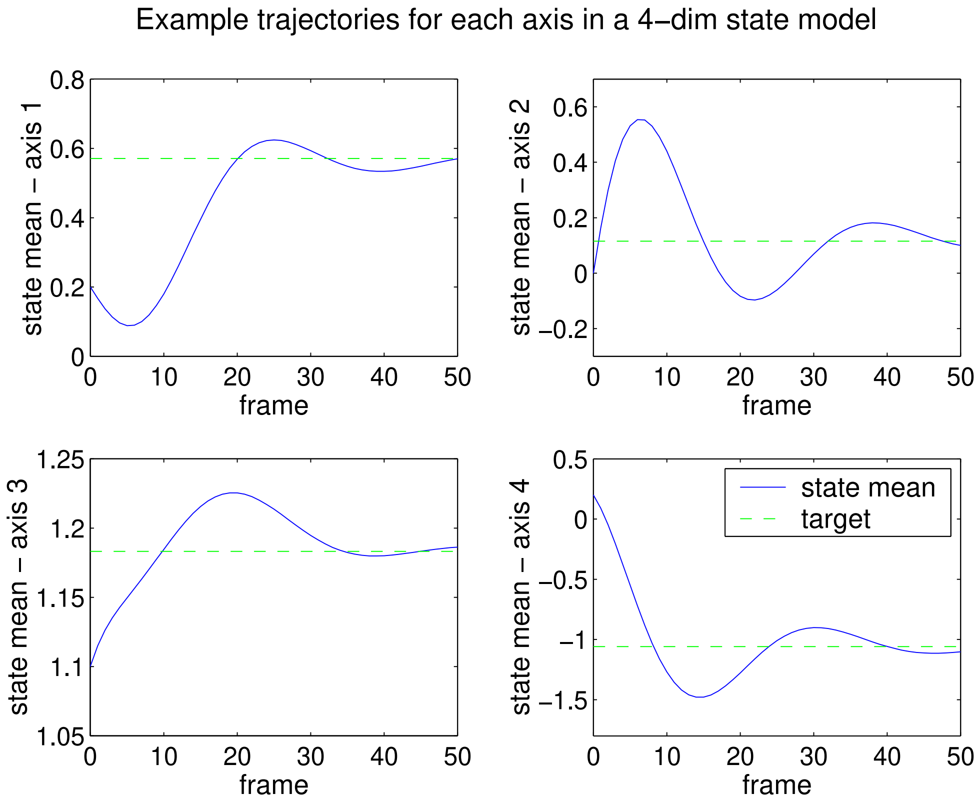

Figure 4.5: These four plots show the model of Figure 4.4 with the state mean target included as a dashed line. Simply used to generate, the state means tend toward their targets.

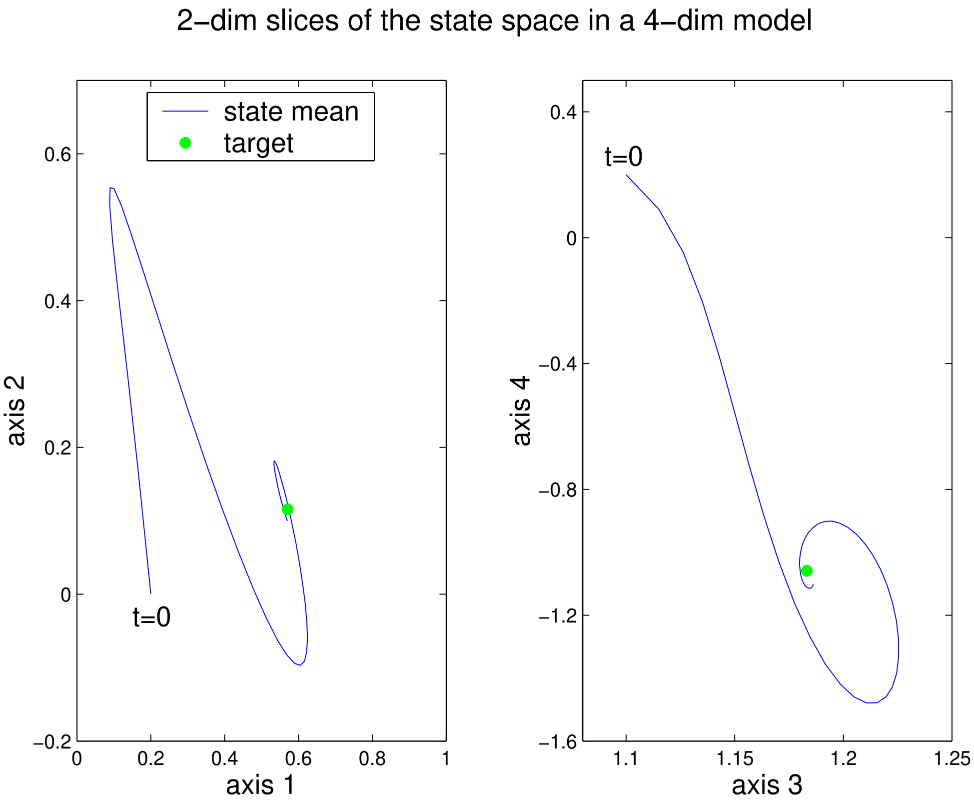

Figure 4.6: The trajectories of Figures 4.5 and 4.4 are shown here as 2-dimensional slices of a 4-dimensional space. Axes 1 and 2 are plotted against each other in the left-hand graph, with 3 and 4 in the right. The state mean targets are shown as red dots.

A further illustration of the state tending toward a target rather than consisting of a fixed set of trajectories is given in Figure 4.7. In each of the four plots, the dash-dotted line corresponds to the 2nd axis of the 4-dimensional state plot in Figure 4.4. The solid line gives the trajectory generated for this axis using the same set of parameters, varying only the state initial mean $\boldsymbol{\pi}$, and once again, the target is shown with a dashed line. Not only do all four trajectories converge rapidly toward the mean target, but also have a common shape within a few frames of generation beginning.

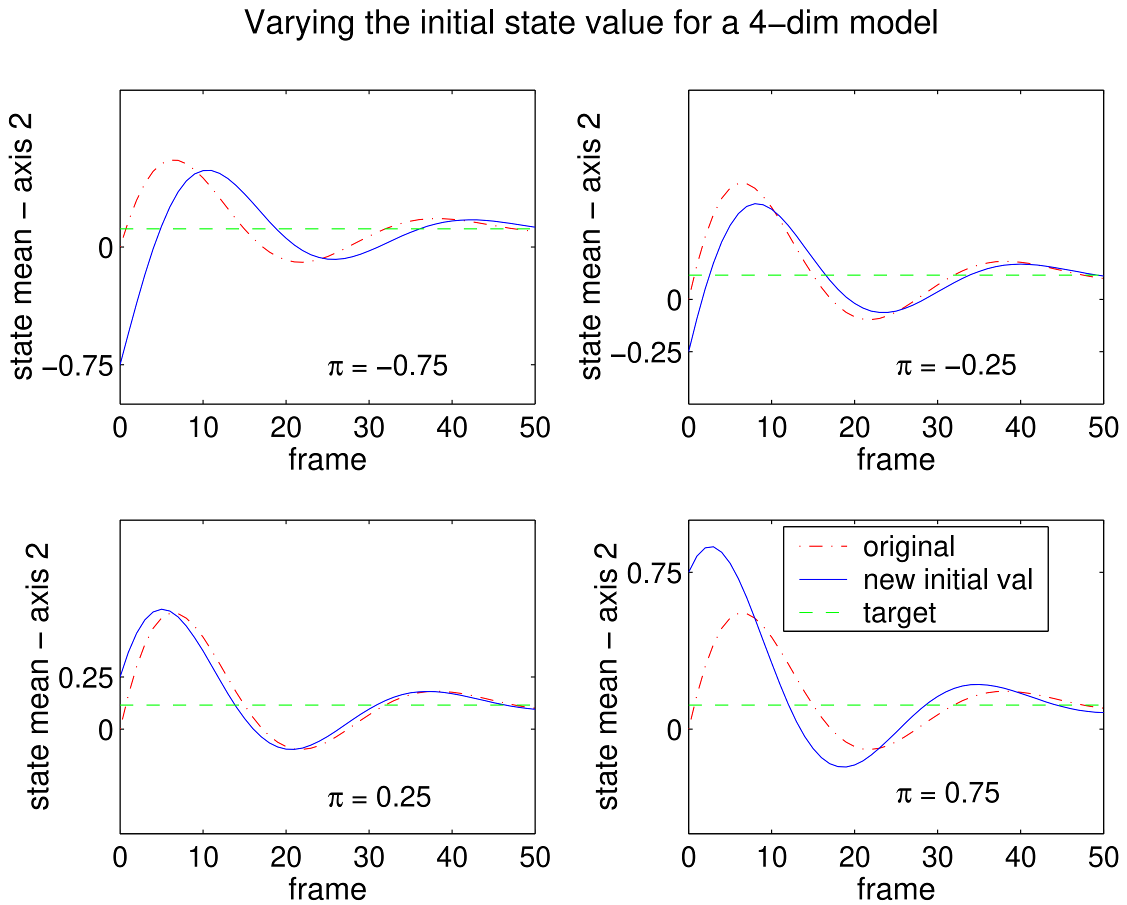

Figure 4.7: The dot-dashed red line in each of these plot corresponds to the second axis of the mean plot in Figure 4.4, and the dashed line its target. The solid line gives state means when the model is used to generate with a variety of initial state values.

State noise

The normally distributed state error, $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$, has two functions. Firstly, a non-zero mean allows the LDM to model a steady drift in the data. The standard formulation often sets $\mathbf{w}$ to zero, and Roweis & Ghahramani (1999) observe that this does not lead to a loss of generality. By adding a $q + 1^{st}$ dimension to the state vector which is always set to unity, an extra column in F to hold $\mathbf{w}$, and filling row $q+1$ with zeros apart from a 1 in the last entry, an exactly equivalent model can be created. Derivations and manipulations of the model can therefore be streamlined by assuming a zero mean on the state error distribution. Whichever way it is incorporated, the state error mean allows the LDM to describe a constant velocity in a given direction. However, this constant displacement will be offset by the decreasing transformation F as the model runs towards its mean target, the location of which is of course affected by $\mathbf{w}$ as shown in Equation 4.12. The second function of the state error $\boldsymbol{\eta}_t$ is that the covariance D corresponds to the intra-segmental variation, as discussed in the introduction to segment models on page 8. This term gives the variance about a given trajectory, and hence the confidence with which the model makes each new prediction.

The properties of the state process have been discussed in this section without reference to the observations. This is a valid exercise since the LDM is a generative model, however in practice, paths through the state space are created by conditioning on observed values. The relevant filtering and smoothing operations will be described after a description of the observation process.

4.1.2 Observation process

The state and observation spaces are linked by a linear transformation and the addition of observation noise. Each dimension of the observation vector is therefore seen as a noisy weighted sum of the state dimensions. As with the state evolution, applying the observation equation can be seen as being composed of two elements. The first is a linear transformation, $\mathbf{y}_t = H\mathbf{x}_t$, which stretches and rotates the state density into a Gaussian distribution over a (usually) higher dimensional space giving:

$$\mathbf{y}_t \sim N(H\mathbf{x}_t,\, H\Sigma_t H^T) \tag{4.13}$$

The second is convolution with the measurement noise $\boldsymbol{\epsilon}_t$ which gives a smoothed and displaced version of 4.13 over the observation space:

$$\mathbf{y}_t \sim N(H\mathbf{x}_t + \mathbf{v}, H\Sigma_t H^T + C) \tag{4.14}$$

As mentioned in Section 2.1.2, the state can be forced to have orthogonal components, in which case modelling of the correlation structure of the data is contained in H.

Observation noise

The observation noise is additive Gaussian noise given by $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$. The mean $\mathbf{v}$ is typically initialised at the start of parameter estimation to be the mean of the observations in the training set. This has the effect of centering the state around the origin and using $\mathbf{v}$ to model the average displacement of the observations. Some statements of the LDM, such as found in Roweis & Ghahramani (1999), assume the data to be zero-meaned and set $\mathbf{v} = \mathbf{0}_p$, where $\mathbf{0}_p$ is a p-dimensional vector of zeros.

Convolving the observation noise with the original predictions to produce Equation 4.14 was described as a smoothing operation. This step widens the spread of the cloud of density over the observations. By shifting probability mass away from regions of high likelihood, the sensitivity to mismatch between training and test data is reduced. Modelling the distribution of the errors between model predictions and the observations in this way corresponds to the extra-segmental variation which was defined in Section 1.3.3 on page 8. The covariance C also gives a measure of the confidence on each prediction the LDM makes, a property which will be examined with reference to articulatory data in Section 4.4.2 on page 108.

4.2 Training and evaluation

The Kalman filter model, as the LDM is also known, has been well used and researched by the engineering and control theory communities since 1960 when Rudolph Kalman introduced his 'new approach to linear filtering and prediction problems' (Kalman 1960). Shortly afterwards, in 1963, Rauch provided the optimal smoother to accompany Kalman's filter (Rauch 1963). With inference possible, researchers turned their attention to parameter estimation, with solutions for H given by Shumway & Stoffer (1982), and for all parameters by Digalakis, Rohlicek & Ostendorf (1993). In this section, these techniques are described, along with other issues and considerations for practical implementation of LDMs.

4.2.1 Inference

The Kalman filter and Rauch-Tung-Striebel (RTS) smoother are used to infer state information given an N-length observation sequence $\mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and a set of model parameters Θ. As a reminder of the terminology introduced in Section 2.1.1 on page 15, filtering is the means of estimating the state distribution at time t given all the observations up to and including that time, $p(\mathbf{x}_t|\mathbf{y}_1^t, \Theta)$. Smoothing gives a corresponding estimate of the state conditioned on the entire observation sequence, $p(\mathbf{x}_t|\mathbf{y}_1^N, \Theta)$. The notation used for the filtered and smoothed state means will be $\hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{x}}_{t|N}$ respectively. The corresponding covariances are written as $\Sigma_{t|t}$ and $\Sigma_{t|N}$.

Kalman filtering takes the initial state distribution[1], $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$ and makes a forward sweep through the observation sequence $\mathbf{y}_1^N$ to produce estimates of $\mathbf{x}_{t|t}$ for $1 \leq t \leq N$. Each recursion consists of two stages. In the first, the model makes prior predictions of the state mean and covariance, $\hat{\mathbf{x}}_{t|t-1}$ and $\Sigma_{t|t-1}$, then in the second, these predictions are projected into the observation space giving $\hat{\mathbf{y}}_t$, compared with $\mathbf{y}_t$, and adjusted to give posteriors, $\hat{\mathbf{x}}_{t|t}$ and $\Sigma_{t|t}$. This process provides a means of updating the state distribution as new observations are made. The adjustment factor K_t is called the Kalman gain and chosen to minimise the filtered state covariance $\Sigma_{t|t}$. The forward filter

[1]The initial state $\mathbf{x}_1$ is one of the parameters which is required to fully specify an LDM. During inference it is used as the prior on the first state vector by setting $\mathbf{x}_{1|0} \sim N(\boldsymbol{\pi}, \Lambda)$.

recursions comprise:

$$
\begin{aligned}
\hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + K_t \mathbf{e}_t \\
\hat{\mathbf{x}}_{t|t-1} &= F\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{w} \\
\mathbf{e}_t &= \mathbf{y}_t - \hat{\mathbf{y}}_t \ = \ \mathbf{y}_t - \mathbf{v} - H\hat{\mathbf{x}}_{t|t-1} \\
K_t &= \Sigma_{t|t-1} H^T \Sigma_{\mathbf{e}_t}^{-1} \\
\Sigma_{\mathbf{e}_t} &= H\Sigma_{t|t-1}H^T + C \\
\Sigma_{t|t} &= \Sigma_{t|t-1} - K_t \Sigma_{\mathbf{e}_t} K_t^T \\
\Sigma_{t,t-1|t} &= (I - K_t H) F \Sigma_{t-1,t-1} \\
\Sigma_{t|t-1} &= F\Sigma_{t-1|t-1}F^T + D
\end{aligned}
$$

This process of predict and correct is shown pictorially in Figure 4.8.

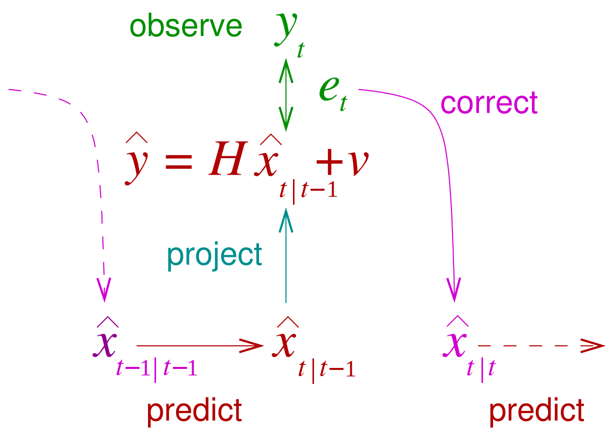

Figure 4.8: This figure shows a single recursion of a Kalman filter. A prediction of $\hat{\mathbf{x}}_{t|t-1}$ is made by the model based on the posterior state for the previous time, $\hat{\mathbf{x}}_{t-1|t-1}$, and projected into the observation space. The error $\mathbf{e}_t$ is then computed with respect to some newly observed data $\mathbf{y}_t$, and the state statistics adjusted to give a posterior $\hat{\mathbf{x}}_{t|t}$

The RTS smoother adds a backward pass in which the state statistics are adjusted once all data has been observed, giving $\hat{\mathbf{x}}_{t|N}$ and $\Sigma_{t|N}$. The RTS smoother can be seen as providing the optimal linear combination of two filters – one which starts at the beginning of the observation sequence and recurses forward, and the other which commences at the final observation and works backward. The weighting of the contribution of each filter is provided by A_t which is chosen to minimise $\Sigma_{t|N}$. The smoother recursions consist of:

$$\begin{aligned}
\hat{\mathbf{x}}_{t-1|N} &= \hat{\mathbf{x}}_{t-1|t-1} + A_t(\hat{\mathbf{x}}_{t|N} - \hat{\mathbf{x}}_{t|t-1}) \\
\Sigma_{t-1|N} &= \Sigma_{t-1|t-1} + A_t(\Sigma_{t|N} - \Sigma_{t|t-1})A_t^T \\
A_t &= \Sigma_{t-1|t-1}F^T\Sigma_{t|t-1}^{-1} \\
\Sigma_{t,t-1|N} &= \Sigma_{t,t-1|t} + (\Sigma_{t|N} - \Sigma_{t|t})\Sigma_{t|t}^{-1}\Sigma_{t,t-1|t}
\end{aligned}$$

The recursions above which estimate the cross-covariance terms $\Sigma_{t,\,t-1|t}$ and $\Sigma_{t,\,t-1|N}$ are not part of the standard filter/smoother equations. However, they are required in parameter estimation for LDMs and are derived in Digalakis et al. (1993), with a more efficient form given by Rosti & Gales (2001).

The plots in Figure 4.9 show a pair of 2-dimensional slices of a 4-dimensional state space during filtering. An LDM was trained on the [`ey`] tokens from a subset of the TIMIT corpus, and then a state sequence generated corresponding to an example of [`ey`] which was not included in the training data. The predict and correct steps are shown, as are the state target and initial value. The target is not attained in either slice, however it can be seen that predictions are typically towards the target, whilst adjustments can be in any direction.

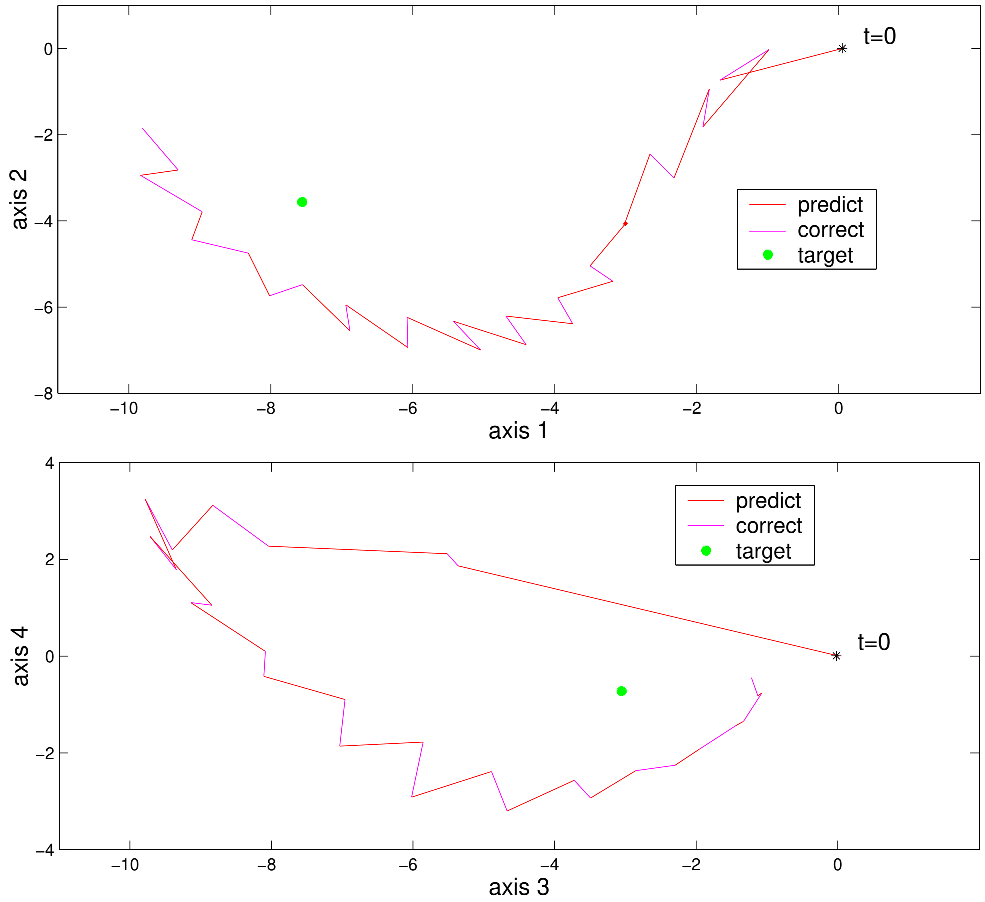

Figure 4.9: An LDM with a 4-dimensional state was trained on the [ey] tokens from a subset of the TIMIT corpus. This figure shows state inference using a Kalman filter given a new unseen [ey] token. Also marked is the state target and initial position. The target is not attained in either slice of the state-space, though trajectories tend toward it.

4.2.2 Parameter estimation

Training an LDM is an *unsupervised* learning problem. There are no inputs to the system, so the model will describe the unconditional density of the observations. As part of the original application of LDMs for speech modelling, Digalakis et al. (1993) present both a classical maximum likelihood approach, and a derivation of the EM algorithm. The latter was adopted for its simplicity and good convergent properties, and is also used in this work. The derivation below largely follows that in Digalakis (1992).

Joint likelihood of state and observations

The state variable and its noise can be combined to give a single Gaussian distributed random variable. Letting Θ denote the model parameter set, from Equation 4.2 we find,

$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\Theta) = \frac{1}{\sqrt{(2\pi)^q|D|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})^T D^{-1}(\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})\right\} \tag{4.15}$$

Similarly from 4.1,

$$p(\mathbf{y}_t|\mathbf{x}_t,\Theta) = \frac{1}{\sqrt{(2\pi)^p|C|}} \exp\left\{-\frac{1}{2}(\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})^T C^{-1}(\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})\right\} \tag{4.16}$$

With $\mathcal{Y} = \mathbf{y}_1^N$ and $\mathcal{X} = \mathbf{x}_1^N$ denoting sequences of observation and state vectors respectively, the Markovian structure of the model means that the joint likelihood of state and observations can be written as:

$$L(\Theta|\mathcal{Y},\mathcal{X}) = p(\mathcal{Y},\mathcal{X}|\Theta) = P(\mathbf{x}_1|\Theta)\prod_{t=2}^{N} P(\mathbf{x}_t|\mathbf{x}_{t-1},\Theta)\prod_{t=1}^{N} P(\mathbf{y}_t|\mathbf{x}_t,\Theta) \tag{4.17}$$

An initial Gaussian density is assumed for the state, and so

$$p(\mathbf{x}_1|\Theta) = \frac{1}{\sqrt{(2\pi)^p|\Lambda|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\pi})^T \Lambda^{-1}(\mathbf{x}_1 - \boldsymbol{\pi})\right\} \tag{4.18}$$

Now substituting 4.15, 4.16, and 4.18 into 4.17, and writing $l(\Theta|\mathcal{Y},\mathcal{X}) = \log L(\Theta|\mathcal{Y},\mathcal{X})$, the joint log-likelihood for the LDM is a sum of quadratic terms:

$$\begin{aligned} l(\Theta|\mathcal{Y},\mathcal{X}) \quad = \quad & -\frac{1}{2}\sum_{t=1}^{N}\left\{\log|C| + (\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})^T C^{-1}(\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})\right\} \\ & -\frac{1}{2}\sum_{t=2}^{N}\left\{\log|D| + (\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})^T D^{-1}(\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})\right\} \\ & -\frac{1}{2}\log|\Lambda| - \frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\pi})^T \Lambda^{-1}(\mathbf{x}_1 - \boldsymbol{\pi}) - \frac{N(p+q)}{2}\log(2\pi) \end{aligned} \tag{4.19}$$

where the last term is due to the normalising constants in 4.15, and 4.16 and 4.18.

Estimation with an observable state

Maximum likelihood parameter estimation involves maximising the log-likelihood function 4.19 for each of the model parameters in turn. This is complicated by the hidden nature of the state, and so it is useful to first consider a slightly altered scenario where the state is actually observed. In this case, true ML estimates of the model parameters can be produced by maximising the log-likelihood function 4.19 for each parameter in turn. This is a question of producing partial derivatives, equating to zero and solving.

Maximising with respect to H proceeds as follows:

$$\begin{aligned} \frac{\partial l}{\partial H} &= \sum_{t=1}^{N} \{C^{-1}\mathbf{y}_t\mathbf{x}_t^T - C^{-1}H\mathbf{x}_t\mathbf{x}_t^T - C^{-1}\mathbf{v}\mathbf{x}_t^T\} = 0 \\ \Rightarrow \hat{H} &= \left(\sum_{t=1}^{N}\mathbf{y}_t\mathbf{x}_t^T - \sum_{t=1}^{N}\hat{\mathbf{v}}\,\mathbf{x}_t^T\right)\left(\sum_{t=1}^{N}\mathbf{x}_t\mathbf{x}_t^T\right)^{-1} \end{aligned} \tag{4.20}$$

and for $\mathbf{v}$:

$$\begin{aligned} \frac{\partial l}{\partial \mathbf{v}} &= \sum_{t=1}^{N} \{C^{-1}\mathbf{y}_t - C^{-1}H\mathbf{x}_t - C^{-1}\mathbf{v}\} = 0 \\ \Rightarrow \hat{\mathbf{v}} &= \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t - \frac{1}{N}\sum_{t=1}^{N}\hat{H}\mathbf{x}_t \end{aligned} \tag{4.21}$$

Multiplying the terms in 4.20 through by $\frac{1}{N}$, 4.20 and 4.21 can be combined to give:

$$\begin{bmatrix} \hat{H} & \hat{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{x}_t^T & \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t \end{bmatrix} \begin{bmatrix} \frac{1}{N}\sum_{t=1}^{N}\mathbf{x}_t\mathbf{x}_t^T & \frac{1}{N}\sum_{t=1}^{N}\mathbf{x}_t \\ \frac{1}{N}\sum_{t=1}^{N}\mathbf{x}_t^T & 1 \end{bmatrix}^{-1}$$

Maximising the log-likelihood function in terms of F and $\mathbf{w}$ follows in an analogous fashion, though summation is from 2 to N, as the initial state distribution is dealt with separately. Finding maximum likelihood solutions for the covariance terms C and D needs a slightly different technique. Noting that maximising with respect to C corresponds to maximising with respect to C^{-1}, and using the result

$$\frac{d}{dZ}\log|Z^k| = kZ^{-T} \tag{4.22}$$

where Z^{-T} denotes the transpose of Z^{-1}, $\hat{C}$ can be found as follows:

$$\begin{aligned}\frac{\partial l}{\partial C^{-1}} &= NC - \sum_{t=1}^{N}(\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})(\mathbf{y}_t - H\mathbf{x}_t - \mathbf{v})^T && (4.23)\\ \Rightarrow \hat{C} &= \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{y}_t^T - \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{x}_t^T\hat{H}^T - \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{v}^T && (4.24)\end{aligned}$$

Maximising in terms of D follows the same line of argument, and solutions for $\boldsymbol{\pi}$ and Λ are also found using the techniques above.

Summary of maximum likelihood solutions

If the state were observable, the maximum likelihood estimates for the parameters of an LDM would be found as:

$$\begin{aligned}
\begin{bmatrix}\hat{H} & \hat{\mathbf{v}}\end{bmatrix} &= \begin{bmatrix}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{x}_t^T & \sum_{t=1}^{N}\mathbf{y}_t\end{bmatrix}\begin{bmatrix}\sum_{t=1}^{N}\mathbf{x}_t\mathbf{x}_t^T & \sum_{t=1}^{N}\mathbf{x}_t\\ \sum_{t=1}^{N}\mathbf{x}_t^T & 1\end{bmatrix}^{-1} && (4.25)\\
\hat{C} &= \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{y}_t^T - \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{x}_t^T\hat{H}^T - \frac{1}{N}\sum_{t=1}^{N}\mathbf{y}_t\mathbf{v}^T && (4.26)\\
\begin{bmatrix}\hat{F} & \hat{\mathbf{w}}\end{bmatrix} &= \begin{bmatrix}\sum_{t=2}^{N}\mathbf{x}_t\mathbf{x}_{t-1}^T & \sum_{t=2}^{N}\mathbf{x}_t\end{bmatrix}\begin{bmatrix}\sum_{t=2}^{N}\mathbf{x}_{t-1}\mathbf{x}_{t-1}^T & \sum_{t=2}^{N}\mathbf{x}_{t-1}\\ \sum_{t=2}^{N}\mathbf{x}_{t-1}^T & 1\end{bmatrix}^{-1} && (4.27)\\
\hat{D} &= \frac{1}{N-1}\sum_{t=2}^{N}\mathbf{x}_t\mathbf{x}_t^T - \frac{1}{N-1}\sum_{t=2}^{N}\mathbf{x}_t\mathbf{x}_{t-1}^T\hat{F}^T - \frac{1}{N-1}\sum_{t=2}^{N}\mathbf{x}_t\mathbf{w}^T && (4.28)\\
\hat{\boldsymbol{\pi}} &= \mathbf{x}_1 && (4.29)\\
\hat{\Lambda} &= \mathbf{x}_1\mathbf{x}_1^T - \mathbf{x}_1\boldsymbol{\pi}^T. && (4.30)
\end{aligned}$$

Estimation with state hidden – application of the EM algorithm

Section 2.1.1 on page 16 described the EM algorithm, which provides a means of iterating toward the ML solution in situations where there is missing or incomplete data. In this case the incomplete data is the state, and EM takes a model with parameters $\Theta^{(i)}$ at the i^{th} iteration and makes an update to give $\Theta^{(i+1)}$, in such a way as to guarantee an increase in the likelihood on the training data.

Dempster et al. (1977) demonstrated that for distributions from the exponential family (of which the LDM with its Gaussian output distribution is a member), the E-step of the EM algorithm consists of computing the conditional expectations of the complete-data sufficient statistics for the standard ML parameter estimates. These sufficient statistics are the quantities in Equations 4.25 – 4.30 which are all computed as sums of

$$\mathbf{y}_t,\ \mathbf{y}_t\mathbf{y}_t^T,\ \mathbf{y}_t\mathbf{x}_t^T,\ \mathbf{x}_t,\ \mathbf{x}_t\mathbf{x}_t^T,\ \mathbf{x}_t\mathbf{x}_{t-1}^T \tag{4.31}$$

Therefore, the E-step involves computing the expectations of the values in 4.31 conditioned on $\mathcal{Y}$ and $\Theta^{(i)}$. Since $\mathcal{Y}$ is observed:

$$\begin{aligned} E[\mathbf{y}_t|\mathcal{Y},\Theta^{(i)}] &= \mathbf{y}_t \\ E[\mathbf{y}_t\mathbf{y}_t^T|\mathcal{Y},\Theta^{(i)}] &= \mathbf{y}_t\mathbf{y}_t^T \\ \text{and}\quad E[\mathbf{y}_t\mathbf{x}_t^T|\mathcal{Y},\Theta^{(i)}] &= \mathbf{y}_t E[\mathbf{x}_t^T|\mathcal{Y},\Theta^{(i)}] \end{aligned}$$

Thus the expectations which must be computed relate to the state $\mathcal{X}$:

$$\begin{gathered} E[\mathbf{x}_t|\mathcal{Y},\Theta^{(i)}] \\ E[\mathbf{x}_t\mathbf{x}_t^T|\mathcal{Y},\Theta^{(i)}] \\ E[\mathbf{x}_t\mathbf{x}_{t-1}^T|\mathcal{Y},\Theta^{(i)}] \end{gathered} \tag{4.32}$$

Given that the initial state $\mathbf{x}_1$ is a normal random variable, and that both state and observation processes are linear with additive Gaussian noise, when conditioned on a sequence of observations $\mathcal{Y}$, the state at time t will also be Gaussian, so

$$\mathbf{x}_t|\mathcal{Y} \sim N(\hat{\mathbf{x}}_{t|N}, \Sigma_{t|N})$$

In this case, with $cov[A,B]$ denoting the covariance of the random variables A and B and using the relation:

$$\begin{aligned} cov[A,B] &= E[AB] - E[A]E[B] \\ \Rightarrow\ E[AB] &= cov[A,B] + E[A]E[B] \end{aligned}$$

the expectations 4.32 can be found as:

$$E[\mathbf{x}_t|\mathcal{Y},\Theta^{(i)}] = \hat{\mathbf{x}}_{t|N} \tag{4.34}$$

$$E[\mathbf{x}_t\mathbf{x}_t^T|\mathcal{Y},\Theta^{(i)}g] = \Sigma_{t|N} + \hat{\mathbf{x}}_{t|N}\hat{\mathbf{x}}_{t|N}^T \tag{4.35}$$

$$E[\mathbf{x}_t\mathbf{x}_{t-1}^T|\mathcal{Y},\Theta^{(i)}] = \Sigma_{t,t-1|N} + \hat{\mathbf{x}}_{t|N}\hat{\mathbf{x}}_{t-1|N}^T \tag{4.36}$$

An RTS smoother as described in Section 4.2.1 on page 89 can be used to compute the complete-data estimates of the state statistics $\hat{\mathbf{x}}_{t|N}$, $\Sigma_{t|N}$, and $\Sigma_{t,t-1|N}$. EM for LDMs then consists of evaluating the ML parameter estimates 4.25 – 4.30 replacing $\mathbf{x}_t$, $\mathbf{x}_t\mathbf{x}_t^T$, and $\mathbf{x}_t\mathbf{x}_{t-1}^T$ with their expectations 4.34 – 4.36.

These solutions easily extend to multiple examples of each time series. In the E-step, the combination of filter and smoother is run for each observation sequence, and sums accumulated over all observations and expected state values. The M-step then proceeds as before by evaluating the expressions 4.25 – 4.30, replacing any divisions by N with a division by the total number of observation frames which contributed to each sum.

EM as presented here is a batch algorithm, meaning that all the training data is processed before the model parameters are updated. A version of EM for on-line learning is given in Neal & Hinton (1998), though the data used to train an ASR system will normally be recorded and annotated at the word or phone level before training commences making a batch approach appropriate.

4.2.3 Likelihood Calculation

Classification and recognition require calculation of the likelihood of a given model generating a section of speech data. The Kalman filter as stated in Section 4.2.1 on page 89 is in a form termed the *innovations representation* by Ljung (1999). The prediction error at time t is given by

$$\begin{aligned} \mathbf{e}_t &= \mathbf{y}_t - \hat{\mathbf{y}}_t \\ &= \mathbf{y}_t - H\hat{\mathbf{x}}_{t|t-1} - \mathbf{v} \end{aligned} \tag{4.37}$$

With the expression for the error $\mathbf{e}_t$ re-written as follows:

$$\begin{aligned} \mathbf{y}_t &= H\mathbf{x}_t + \boldsymbol{\epsilon}_t & (4.38)\\ \Rightarrow \mathbf{y}_t - H\hat{\mathbf{x}}_{t|t-1} - \mathbf{v} &= H(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) - \mathbf{v} + \boldsymbol{\epsilon}_t & (4.39)\\ \Rightarrow \mathbf{e}_t &= H(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + (\boldsymbol{\epsilon}_t - \mathbf{v}) & (4.40) \end{aligned}$$

the associated covariance is given as:

$$\begin{aligned} \Sigma_{\mathbf{e}_t} &= E[\mathbf{e}_t\, \mathbf{e}_t^T] \\ &= E\left[\left(H(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + (\boldsymbol{\epsilon}_t - \mathbf{v})\right)\left(H(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + (\boldsymbol{\epsilon}_t - \mathbf{v})\right)^T\right] & (4.41)\\ &= E\left[H(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})^T H^T\right] + E\left[(\boldsymbol{\epsilon}_t - \mathbf{v})(\boldsymbol{\epsilon}_t - \mathbf{v})^T\right] & (4.42)\\ &= H\Sigma_{t|t-1}H^T + C & (4.43) \end{aligned}$$

Since errors are assumed uncorrelated and Gaussian, the log-likelihood of an observed sequence $\mathbf{y}_1^N$ given an LDM with parameter set Θ can be calculated as:

$$log\, p(\mathbf{y}_1^N|\Theta) = -\frac{1}{2}\sum_1^N \left\{log|\Sigma_{\mathbf{e}_t}| + \mathbf{e}_t^T\, \Sigma_{\mathbf{e}_t}^{-1}\, \mathbf{e}_t\right\} - \frac{Np}{2}\log(2\pi) \qquad (4.44)$$

where $\mathbf{e}_t$ and $\Sigma_{\mathbf{e}_t}$ are computed as part of the standard Kalman filter recursions. The normalisation term outside the summation in 4.44 can be omitted when comparing multiple models on a single given section of data as occurs during classification of a single segment or recognition of a single utterance.

It was found by experiment that the state's contribution to the error covariance $\Sigma_{\mathbf{e}_t}$ was detrimental to classification performance. The state covariance is normally reset to a value learned during training at the start of each segment, and converges during the first few Kalman filter recursions. The resulting fluctuations in the likelihoods computed during the segment-initial frames have most effect on the overall likelihood of shorter phone segments. Replacing $\Sigma_{\mathbf{e}_t} = C + H\Sigma_{t|t-1}H^T$ with $\Sigma'_{\mathbf{e}_t} = C$ improves classification accuracy on shorter segments. This is demonstrated in the results of phone classification on 480 TIMIT validation sentences where an LDM has been trained on the data corresponding to each phone class. Figure 4.10 shows, for each segment length in frames, the number of correctly classified tokens. It is apparent that for segments over 11 frames, the correct form of likelihood calculation gives a marginally improved accuracy. However, for

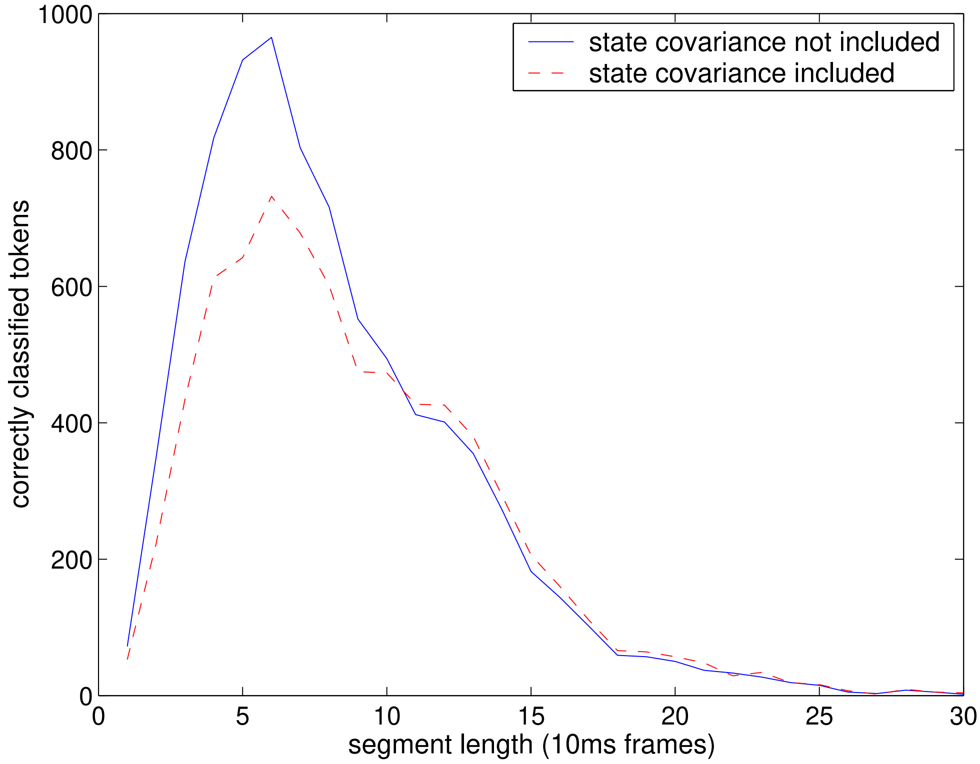

Figure 4.10: Results of phone classification on 480 TIMIT validation sentences, broken down by segment length. The dashed red line corresponds to classification using the correct form of likelihood calculation, and the solid blue line to classification where likelihoods are computed replacing $\Sigma_{\mathbf{e}_t} = C + H\Sigma_{t|t-1}H^T$ with $\Sigma'_{\mathbf{e}_t} = C$.

shorter segments, a modified $\Sigma_{\mathbf{e}_t}$ gives markedly higher classification performance. The results shown are on for the 61 phone TIMIT set, prior to the addition of language model. On these 480 sentences, using the correct and modified likelihood calculations results in classification accuracies of 40.1% and 46.7% respectively. Likelihood calculations for the experiments in this thesis will omit the contribution of the state covariance unless otherwise stated. Further discussion of the properties of the state covariance are found in Section 7.1.4 on page 224.

4.2.4 Implementational Issues

Efficient computation The initial distribution of the state, $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$, is part of the specification of an LDM. These values are estimated during training, and then used to initialise the state priors for the Kalman filter, so that $\mathbf{x}_{1|0} = \boldsymbol{\pi}$ and $\Sigma_{1|0} = \Lambda$. An examination of the filter and smoother recursion given on pages 90 and 91 reveals that the none of computations for the 2^{nd} order statistics at time t involve the newly observed value $\mathbf{y}_t$. The forward statistics $\Sigma_{t|t-1}$, $\Sigma_{t|t}$, $\Sigma_{t,t-1|t}$, K_t, and $\Sigma_{\mathbf{e}\,t}$ will then be identical for any pair of observation sequences $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_1}\}$ and $\{\mathbf{y}'_1, \ldots, \mathbf{y}'_{N_2}\}$ for $t \leq N_1, N_2$.

The situation is slightly different for the backward smoothing pass, though the above also applies to A_t, which is calculated using the filtered parameters $\Sigma_{t-1|t-1}$ and $\Sigma_{t|t-1}$. However, the smoothed state covariances are dependent on N, and so $\Sigma_{t-1|N}$ and $\Sigma_{t,t-1|N}$ are identical for any pair of observation sequences $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_1}\}$ and $\{\mathbf{y}'_1, \ldots, \mathbf{y}'_{N_2}\}$ for which $N_1 = N_2$. These observations lead to implementational strategies in which state covariances and the correction factors K_t and A_t can be calculated, cached, and reused. The matrix operations which are used to compute these quantities form the bulk of the computation of implementing LDMs and so considerable speed-ups can be found by employing such a strategy.

For example, training a full set of LDMs for a single iteration of EM on 12 MFCCs and energy derived from the TIMIT training data on a 2.4GHz Pentium P4 processor took 10 minutes, 39 seconds. This was reduced to 1 minute 54 seconds through caching. Similarly, using those same models for classification of the full TIMIT test set took 108 minutes 32 seconds, reduced to 7 minutes 39 seconds by pre-computing the relevant quantities. This represents a 14-fold speed increase.

Constraints Taking an LDM and multiplying one dimension of the state space by some factor whilst dividing the corresponding column of H by the same gives distributions over the observations identical to those of the original. Despite the lack of unique parameter estimates, and the inherent degeneracy which was discussed in Section 2.1.2 on page 20, EM training for LDMs is stable in practice and converges quickly. As with any application of EM, parameters must be initialised before training begins. There is no single established or 'correct' technique for initialising LDMs, though choice of initial parameters is key to good performance. Appendix B describes the approach used in this work.

One constraint is always placed on the LDMs during training, which is that F is a decaying mapping. If $|F| > 1$ were allowed, the state evolution could give a model of exponential growth. Such behaviour may not be apparent over small numbers of frames, whilst still introducing an element of numerical instability into the system. This becomes especially important in the situation where the state is not reset between models. To constrain $|F| < 1$, the singular value decomposition (SVD) is used at the re-estimation step. The SVD provides a pair of orthonormal bases U and V, and a diagonal matrix of singular values S such that

$$F = USV^T \tag{4.45}$$

By replacing any elements of S greater than $1-\epsilon$ with $1-\epsilon$ for some small ϵ ($\epsilon = 0.005$ was used in this work), and then re-computing $F = US_{new}V^T$, the bases of F are preserved whilst forcing the transform along them to be decaying (Roweis 2001).

Other constraints may be considered also. These include forcing the rows or columns of H to sum to unity in order to fix the scaling of the state process. Alternatively, one or both of C and D can be set to be diagonal, thereby forcing modelling of the correlation structure of the data into H, and using the error distributions to describe the variances unique to each dimension of the data. The latter can be enforced during the re-estimation step, simply by setting all off-diagonal elements of $\hat{C}$ or $\hat{D}$ to zero. This would also increase implementational efficiency, as inverting diagonal matrices requires far less computation than their full counterparts.

4.3 A comparison of LDMs and autoregressive processes

A vector autoregressive model describing an N-length sequence of p-dimensional random variables $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ can be written as:

$$\mathbf{z}_t = \sum_{i=1}^{r} A_i \mathbf{z}_{t-i} + \boldsymbol{\eta}_t \tag{4.46}$$

where the A_is are $p \times p$ matrices and $\boldsymbol{\eta}_t$ is additive Gaussian noise given by $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$. Williams (2003) shows that the likelihood of observations $\mathcal{Z}$ under such a model can be made equivalent to a likelihood expressed in terms of independent difference observations derived from $\mathcal{Z}$. This result demonstrates that an autoregressive model can be expressed as a static model with an appropriate set of δ coefficients. The question follows as to whether there is then an equivalence between an LDM and a static model with δ coefficients.

Firstly, note that the model of Equation 4.46 can be written in the following form:

$$\mathbf{Z}_{t-r+1}^{t} = \begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \\ \vdots \\ \mathbf{z}_{t-r+1} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & \cdots & & A_r \\ I_p & 0_p & \cdots & & 0_p \\ 0_p & I_p & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0_p & \cdots & 0_p & I_p & 0_p \end{bmatrix} \begin{bmatrix} \mathbf{z}_{t-1} \\ \mathbf{z}_{t-2} \\ \vdots \\ \mathbf{z}_{t-r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \end{bmatrix} \tag{4.47}$$

where I_p represents a $p \times p$ identity matrix, 0_p a $p \times p$ matrix of zeros, and $\mathbf{0}_p$ a vector of zeros length p. Then letting $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ be a set of p-dimensional observations which we wish to model with the relationship $\mathbf{y}_t = \mathbf{z}_t + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$, we can write:

$$\mathbf{y}_t = \begin{bmatrix} I_p & 0_p & \cdots & 0_p \end{bmatrix} \begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \\ \vdots \\ \mathbf{z}_{t-r+1} \end{bmatrix} + \boldsymbol{\epsilon}_t \tag{4.48}$$

$$\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \\ \vdots \\ \mathbf{z}_{t-r+1} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & \cdots & & A_r \\ I_p & 0_p & \cdots & & 0_p \\ 0_p & I_p & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0_p & \cdots & 0_p & I_p & 0_p \end{bmatrix} \begin{bmatrix} \mathbf{z}_{t-1} \\ \mathbf{z}_{t-2} \\ \vdots \\ \mathbf{z}_{t-r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \end{bmatrix} \tag{4.49}$$

By setting the covariance C of $\boldsymbol{\epsilon}_t$ to be zero, the model remains a vector autoregressive process with $\mathcal{Y}$ simply a displaced version of $\mathcal{Z}$. However, the addition of (observation) noise through $\boldsymbol{\epsilon}_t$ renders $\mathcal{Z}$ a hidden variable, and makes the model described by Equations 4.48 and 4.49 into an LDM that has been subjected to a number of constraints. The evolution matrix in Equation 4.49 holds the original r-order autoregressive process as well as acting as a shift operator for each component of the stacked hidden state vector $\mathbf{Z}_{t-r+1}^{t}$. Writing the LDM in this form shows how the state can 'remember' past values, which here are noisy versions of the observations.

So far, $\mathcal{Z}$ has been of the same dimension as the observations $\mathcal{Y}$, which means that the hidden state vector has been of dimension rp. State-space models generally employ a state of different (frequently lower) dimension than the observations. Incorporating dimensionality reduction via the observation process means that the autoregressive model can have just as many degrees of freedom as required to model any dynamics which might underly the observations. Now letting $\mathcal{Z}$ be a d-dimensional vector, and with the B_is representing $p \times d$ matrices, Equations 4.48 and 4.49 can be written as:

$$\mathbf{y}_t = \begin{bmatrix} B_1 & B_2 & \cdots & B_r \end{bmatrix} \mathbf{Z}_{t-r+1}^{t} + \boldsymbol{\epsilon}_t \tag{4.50}$$

$$\mathbf{Z}_{t-r+1}^{t} = \begin{bmatrix} A_1 & A_2 & & \cdots & A_r \\ I_d & 0_d & & \cdots & 0_d \\ 0_d & I_d & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0_d & \cdots & 0_d & I_d & 0_d \end{bmatrix} \mathbf{Z}_{t-r}^{t-1} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \mathbf{0}_d \\ \vdots \\ \mathbf{0}_d \end{bmatrix} \tag{4.51}$$

Note that the model of equations 4.48 and 4.49 can be found by setting $p = d$, $B_1 = I_d$ and $B_i = 0_d$ for $i = \{2, \ldots, r\}$.

The matrices B_i can be used to incorporate linear dimensionality reduction into the model. Specifying B_1 but setting the remaining B_is to be zero matrices ensures that y_t has a dependence only on z_t. In this case, the observations are modelled as a corrupted-by-noise version of a lower-dimensional autoregressive process of order r. Further specifying B_i for $i = \{2, \ldots, r\}$ gives y_t a dependence also on z_{t-i}.

In practice, estimation for LDMs is largely unconstrained. The state vector is not explicitly divided into separate components, as rd-dimensional $\mathbf{Z}_{t-r+1}^{t}$ is replaced by q-dimensional $\mathbf{x}_t$. Neither of the state evolution or observation matrices, F and H respectively, are forced to place zeros as shown in Equations 4.48 and 4.49. Writing the model in this fashion simply serves to show the sorts of structure which, subject to appropriate estimation techniques, the LDM might discover in the data.

This interpretation of the modelling of the LDM aims to highlight the differences between LDMs and autoregressive models. The addition of observation noise sets the two apart by making the autoregressive component into a hidden process. When combined with dimensionality reduction via the observation process, the effect is to ambiguate the order of the modelling in the state. The equivalence between an autoregressive model and a static model with δ features is due to the explicit linear relationship an autoregressive process describes between observed feature vectors. This section has shown how modelling is altered and hence how this explicit relationship is removed when the autoregressive process instead describes an internal state of the model.

4.4 The LDM as a model for speech recognition

A straightforward application of LDMs to acoustic modelling for ASR is to train a single model for each phone class in the inventory of a given corpus. This is the case in most of the experiments which will be presented in this thesis. This will be referred to as the *LDM-of-phone* formulation. Using a single LDM to characterise variable-length phone-segments in this way makes the following assumptions:

- the correlation between consecutive frames is constant. The relationship between every pair of consecutive frames is identical within each segment.
- segments are not duration-normalised. Therefore, shorter instances of phones are treated as having the same dynamic characteristics as some portion of a longer phone.
- the output distribution can be described by a single full-covariance Gaussian distribution. However, the mean and covariance are subject to systematic variation throughout any given segment.

The internal variables in the hidden state reflect some of the known properties of speech production, where articulators move relatively slowly along constrained trajectories. The continuous nature of the state means that temporal dependencies are modelled for as long as the state is not reset, with the position of the state at time t affecting its position at time $t + \tau$. This could be the beginning and end of a phone or sentence depending on how the model is implemented.

4.4.1 Describing acoustic data

The model has built into it the notion of speech being modelled in a domain other than the observations, which are seen as noisy transforms of an underlying process. A linear mapping between state and observation processes dictates that points which are close in state space are also close in observation space. Therefore, trajectories which are continuous in state space are also continuous in observation space. If the hidden state is seen as having articulator-like characteristics, such a constraint is not universally appropriate as sometimes small changes in articulatory configuration can lead to radical changes in the acoustics (examples of this were given in Section 3.1.1 on page 50). However, Section 3.4 on page 66 showed that whilst linear models do not give good descriptions of the dependencies between phone segments, behaviour *within* phones can be adequately accounted for by a linear predictor. This is reflected in the LDM-of-phone formulation: within phone models, the output distribution evolves in a linear, continuous fashion. Discontinuities and non-linearities can be incorporated at phone boundaries where resetting the state and switching the observation process parameters H, $\mathbf{v}$ and D results in a sudden shift in acoustic space. Alternatively, by passing state statistics across model boundaries, the state process can remain continuous through such shifts.

Figures 4.11 and 4.12 give a visual depiction of how well an LDM can characterise acoustic data. The top spectrogram was generated from the actual cepstral coefficients, and is plotted as time against log frequency for the TIMIT sentence, 'Do atypical farmers grow oats?' Red corresponds to regions of high energy, and blue to low. The second spectrogram represents predictions of the cepstra made by a series of LDMs. A set of phone models trained on the TIMIT training data was used to generate predictions the length of the utterance. The time-aligned phonetic labels dictated which model was run in each phone region, and the state predictions made during a forward pass of a Kalman filter were transformed by the observation process to give a vector of mean predicted cepstra for each frame. The state statistics were reset at the beginning of each new phone to model-specific values, learnt during training. The LDM follows many of the spectral characteristics present in the original spectrogram, though the effect of resetting the state statistics at phone boundaries is apparent.

These spectrograms illustrate the importance of modelling spectral dynamics. Some phones, such as the diphthong [`ey`], 5^{th} segment in the utterance, are characterised by the manner in which the vocal tract transfer function, and hence the frequency patterns of the speech signal, change over time. A static model would not be able to generate data with these characteristics.

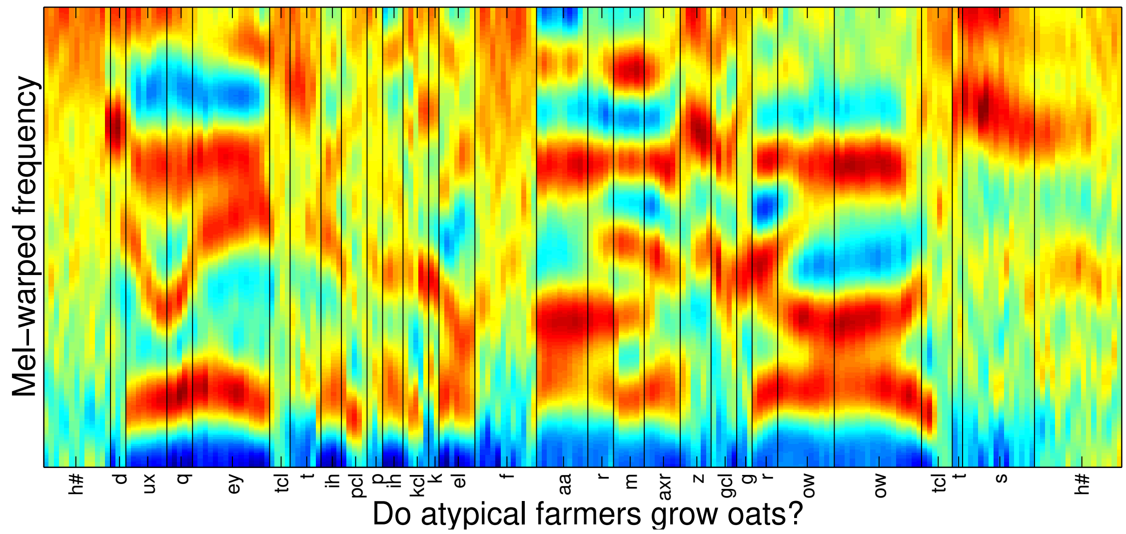

Figure 4.11: A spectrogram generated from the actual Mel-cepstra for the TIMIT sentence 'Do atypical farmers grow oats?'

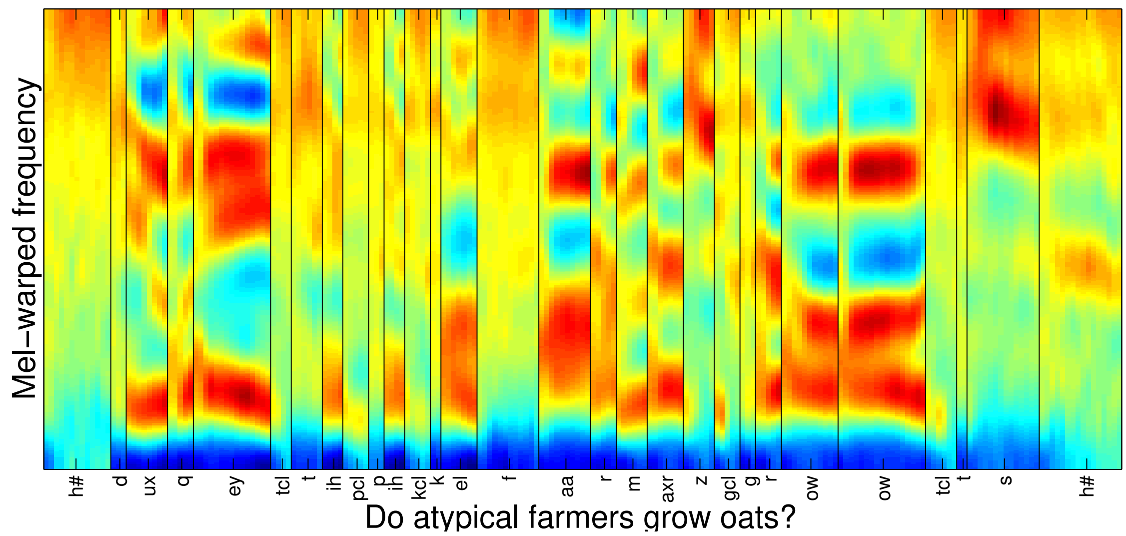

Figure 4.12: A spectrogram generated from predictions made by an LDMs during a forward pass through the data of the same TIMIT sentence as above. Many of the same features are captured in the LDM predictions, though the effect of resetting the state statistics at phone boundaries is apparent

4.4.2 Describing articulatory parameters

Measured articulatory parameters have many of the same properties as the trajectories which LDMs can generate, being smooth, slowly varying and continuous, yet noisy. Furthermore, the model can absorb the asynchrony which exists between the motions of different articulators. Making a model of measured articulatory data, such as that found in the MOCHA corpus, is a situation in which fewer degrees of freedom will most likely be required for modelling purposes than are originally present in the data. For example, the EMA data includes x and y coordinates for three points on the tongue. These six data streams are likely to be highly correlated and there will be redundancy of information. Subspace modelling should be able provide a compact representation of such a system.

The observation noise C has an interesting interpretation when dealing with articulatory data, as it can be seen as capturing the critical, or otherwise, nature of an articulator during a given phone (see page 51 for a definition of critical articulators). It would be expected that the variances found on the data stream corresponding to a critical articulator will be low compared to that of an articulator which is not critical for a given phone.

Work reported in Richmond et al. (2003) was concerned with recovering articulatory traces from acoustic parameters using data from the MOCHA corpus. The variances associated with recovery of articulator feature dimensions tongue tip y, upper lip x and velum x were compared across the consonantal phones. The nature of the EMA data is such that y-coordinates correspond to height and x-coordinates to frontness[2]. In each case, the phones for which the variance was low were found to correspond to those in which the articulator was expected to be critical. Likewise, those with a high variance on the network output corresponded to the articulators thought to be non-critical for production of the given phone.

A similar experiment was conducted as part of this thesis' exploration of LDMs. Models were trained on the EMA data for all 460 sentences from speaker `fsew0` of the MOCHA corpus, and the diagonal elements of the observation noise covariance C corresponding to the three articulatory feature dimensions above were extracted. Table 4.1 gives the variances for the 23 consonants in the MOCHA phone set ranked in order of magnitude.

[2]This is approximate as x and y are relative to the bite-plane, which is measured during EMA recording.

The results tend to follow what might be expected, with tongue tip being critical for production of [sh,s,z], and upper lip being non-critical for [y,t,d]. Furthermore, the variance associated with nasalised phones for which the velum is lowered and open [m,ng,n] is high compared to when it is closed, such as [zh,ch,jh]. One notable exception is for [t], which gives one the largest variances for the tongue tip, an articulator which would be expected to be critical.

tongue tip y		upper lip x		velum x	
phone	variance	phone	variance	phone	variance
[sh]	0.012	[m]	0.038	[zh]	0.014
[s]	0.012	[w]	0.039	[ch]	0.019
[z]	0.013	[th]	0.040	[jh]	0.023
[zh]	0.014	[b]	0.040	[b]	0.024
[th]	0.015	[p]	0.040	[sh]	0.029
[jh]	0.021	[ch]	0.040	[th]	0.029
[dh]	0.023	[f]	0.046	[p]	0.031
[y]	0.029	[zh]	0.046	[z]	0.031
[ng]	0.030	[sh]	0.050	[w]	0.032
[n]	0.030	[z]	0.051	[f]	0.033
[k]	0.031	[v]	0.052	[s]	0.033
[p]	0.032	[ng]	0.053	[v]	0.035
[d]	0.033	[s]	0.054	[y]	0.038
[v]	0.036	[jh]	0.054	[l]	0.039
[ch]	0.037	[l]	0.057	[g]	0.040
[f]	0.037	[n]	0.057	[t]	0.043
[g]	0.038	[dh]	0.060	[d]	0.046
[m]	0.039	[g]	0.062	[dh]	0.054
[w]	0.041	[r]	0.063	[k]	0.054
[b]	0.042	[k]	0.063	[r]	0.058
[t]	0.046	[y]	0.065	[m]	0.062
[l]	0.055	[t]	0.065	[ng]	0.071
[r]	0.059	[d]	0.071	[n]	0.073

Table 4.1: LDMs were trained on the MOCHA EMA data, and shown here for the 23 consonantal phones are the ranked variances associated with articulatory dimensions tongue tip y, upper lip x and velum x. y-coordinates correspond to height and x-coordinates to frontness. In general, low variances correspond to phones in which the articulator would be expected to be critical.

4.4.3 Points for investigation

Since the use of LDMs for speech recognition is largely uncharted, many aspects of its application are open for investigation. The task of classification is the ideal domain in which to compare experimentally a number of modelling alternatives, which are described below.

Modelling dynamics The question of central importance is whether a first-order linear model of the underlying dynamics is useful for discriminating phones. It is this potential to model the correlations between consecutive frames which sets the LDM apart from a straightforward Gaussian classifier. The contribution of the dynamic portion of the model can be investigated through comparison with models with otherwise equal modelling power. Setting the state to be dimension zero, or equivalently $H = 0$, gives a straightforward Gaussian classifier, as all modelling is through the observation noise, $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$. Alternatively, a factor analyser model, as described in Section 2.1.2 on page 19, sets $F = 0$ and can be seen as an LDM without state dynamics. The factor analyser shares subspace modelling and EM training with the LDM, though differs by making no model of the inter-frame dependencies. Comparing the classification performance of LDMs with that of factor analysers gives a useful indication of the contribution of state dynamics.

State dimension The discussion of the state process above illustrated the effect the dimension of the hidden state has on the complexity of the motions which can be modelled. If a continuous, dynamic state-space has something to offer, how many degrees of freedom in the state are needed for speech data?

Form of H The original application of LDMs to speech as part of the SSM described in Section 2.3.6 set H to the identity matrix. The model was thus cast as a smoothed Gauss-Markov model, and subspace modelling was ignored. If there are fewer degrees of freedom present in the data than in the observation, the compact parameterization offered by a sub-space should improve modelling.

Form of the error terms The effect of forcing the error covariances to be diagonal was discussed above. Forcing D to be diagonal or an identity matrix gives no theoretical loss of generality, and can be implemented with one of two methods. An equivalent model can be created by subsuming the correlation structure in to F and H leaving $D = I$, or alternatively off-diagonal components can be set to zero during the M-step of re-estimation. The latter may affect performance as correlation information accumulated during the E-step is simply ignored. It may be useful to stop training the observation noise $\boldsymbol{\epsilon}_t$ after a few iterations, to focus any learning on to the other parameters.

Chapter 5

LDMs for Classification of Speech

Using LDMs for full speech recognition is of course the ultimate goal of this work. However, a classification task is an extremely useful staging post. Whereas recognition involves jointly finding the most likely model sequence and segmentation, in a classification task the segment start and end times are given and it is only the model sequence which must be determined. This provides a framework in which to make comparisons between models where the number of confounding factors is kept to a minimum. The experimenter is able to refine the process of parameter initialisation, training and testing, safe in the knowledge that no errors are introduced from such sources as decoding or duration modelling.

Sections 5.1 and 5.2 present the results of speaker-dependent MOCHA and speaker-independent TIMIT phone classification tasks respectively. The experimental set-up is straightforward, and the intention is to compare classification performance under a variety of models and parameterizations of speech. Section 5.3 extends these basic results by looking at a number of ways in which to develop the acoustic modelling.

In the classification experiments which follow, the LDMs are fully specified. The usefulness or otherwise of the dynamic component of the model is what is being assessed, and will be decided on the relative performance of otherwise equivalent static models, multivariate Gaussians and factor analysers (FA), compared with LDMs. As discussed in Section 2.1.2 on page 19, factor analysers produce spatially correlated Gaussian output distributions but with substantially fewer parameters than present in a full covariance Gaussian. This may prove advantageous if there is insufficient data to give robust estimates of the covariance matrices of standard Gaussian distributions. These experiments will also enable comparison of model performance on a range of acoustic, articulatory and combined feature sets. Testing of other assumptions relating to LDM formulation will follow in Section 5.2.4.

Paired t-test for comparison of results

A paired t-test will be used to aid comparison of experimental set-ups. Such a test provides a method for assessing if system A has given a *consistently* higher accuracy than system B across the test-set. The test sentences are split into n groups (where $n = 10, 24$ for MOCHA, TIMIT data respectively), and the classification accuracies under both systems computed for each group. The hypothesis that the mean accuracy difference $\bar{d}$ between the systems is 0, $H_0 : \bar{d} = 0$ is tested against the one-sided alternative $H_1 : \bar{d} > 0$ by computing

$$t = \frac{\bar{d}}{\sqrt{s_d^2/n}} \tag{5.1}$$

where s_d^2 is the sample variance of the differences and n the number of pairs. This is compared to a t-distribution with $n-1$ degrees of freedom to give the probability of finding such a value of t by chance. Low probabilities ($p < 0.05$ or $p < 0.01$) justify rejecting H_0 in favour of H_1 and concluding that there is evidence supporting system A's superior performance. In this work, $p < 0.01$ will be assumed as a threshold unless otherwise stated.

For a number of the MOCHA experiments, K-fold cross-validation as described below in section 5.1.1 is used. Such a scheme is necessary given the small size of the MOCHA corpus, though can lead to problems when comparing classification results found under competing models. This comes about as no account of the variation due to the training set is taken. This leads to underestimates of the error about the overall accuracy, and a tendency therefore to find differences statistically significant. However, in the experiments which follow, this problem is avoided by making comparisons between pairs which have matching training and test divisions under the cross-validation scheme (Bengio & Granvalet 2003).

5.1 Speaker-dependent MOCHA classification

5.1.1 Methodology

In all experiments, the data is split into three subsets, each of which has a distinct role. Some is used for parameter estimation (training set), some for making intermediate decisions (validation set), and the remainder is used for evaluation (test set). Models are trained using an application of the EM algorithm, which, as with any iterative estimation procedure can lead to overfitting of the data. In this case, the models learn the specific characteristics of the training set but do not generalize well and perform poorly on unseen data. Using a validation set provides a means of setting parameters such as the number of training iterations and the language model scaling factor before the final models are evaluated on the test set.

Training The time-aligned labels provided with the MOCHA corpus are used to extract all tokens corresponding to each of the 46 phone classes, so that a single context-independent model can be trained for each. A number of EM iterations are carried out with the parameters being stored at each. The bigram language model probabilities are also estimated using the phone pairs present in the training set.

Evaluation Model likelihoods are evaluated by summing normalised prediction error log-likelihoods as described in Section 4.2.3 on page 97. With the labels providing start and end times for each segment, a log-likelihood can be computed under each of the 46 models for all segments in an unseen utterance. A Viterbi search with a simple bigram language model (no backing off is used) then provides a means of choosing the most likely phone sequence. The acoustic likelihoods are not normalised, so an appropriate scaling of the language model log probability must be found. The combination of scaling factor and model parameter estimation iterations which gives the highest accuracy on the validation set is used for a final evaluation on the test data. Between 4 and 6 EM iterations and language model scaling factors of between 10 and 20 typically give the highest accuracies.

Basic classification procedure

To ensure enough data for robust estimation, 4/5 of the 460 utterances are set aside for training, and the remaining 1/5 divided equally between validation and test sets. Such a division leaves quite a small test set and so, where possible, main classification results are found using a K-fold cross-validation procedure which is described below. As long as a test set consisting of only 46 sentences reflects the properties of the entire data, it can be used to find preliminary results and the computation involved will only be a fraction of full cross-validation. Results on the small test-set will be compared to those found with a full cross-validation below on page 119.

K-fold cross-validation procedure

The data is split into 5 equally-sized cross-validation sets A, B, C, D, and E. By rotating the function of the data in each set (training, validation, and testing), all utterances in the corpus are at some stage used for final evaluation. The full scheme proceeds as follows:

- models trained on 3 out of 5 sets, initially A, B and C.
  - model parameters at each training iteration stored
- validation performed on a 4^{th} set, D.
  - best set of models and language model scaling factor chosen
- models retrained from scratch on original 3 plus 4^{th} cross-validation sets, A, B, C and D
- models tested on the 5^{th}, as yet unused set, E.

This process is repeated 5 times, with A, B, C, D and E switching roles until each has been used as a test-set. Table 5.1.1 shows the permutations used to enable a final classification accuracy to be calculated for all the utterances in the corpus. From now on, classification will refer to the basic classification procedure with the 46 utterance test set, and when cross-validation has been used, it will be stated explicitly. Note that where results are compared, evaluation will always be given for identical test sets.

train set	validation set	test set
A, B, C	D	E
B, C, D	E	A
C, D, E	A	B
D, E, A	B	C
E, A, B	C	D

Table 5.1: The 5 cross-validation sets swap role until each has been used for testing

Reminder of the MOCHA features

The data-sets which are derived from the MOCHA corpus were described in Chapter 3. Table 3.1 is repeated here as Table 5.2 to give a reminder of the various features which will be used for experimentation. Where feature dimensions are 46 or greater, additional sets are included where linear discriminant analysis (LDA) has been applied for dimensionality reduction. No experiments are carried out without LDA post-processing for the combined data sets with δ and $\delta\delta$ parameters as the resulting 81-dimensional features are considered excessively large.

Roadmap for the experiments which follow

As stated in the introduction to this chapter on page 113, the main thrust of this set of experiments is to examine the contribution a dynamic hidden state makes to phone classification accuracy, compared with otherwise equivalent static models. Results are presented for each of the feature sets above, starting with MOCHA articulatory features in section 5.1.3 on page 122 and acoustic-derived features in Section 5.1.4 on page 132. Sections 5.1.5 and 5.1.6 on pages 137 and 143 then give classification results where acoustic features are combined with real and network-recovered EMA respectively. Section 5.1.7 on page 148 presents a summary of the findings thus far.

To begin with, Section 5.1.2 describes preliminary experiments which compare results on the small MOCHA test set and those found using a full K-fold cross-validation, and also look at the effect that the data frame-rate has on classification accuracy.

type	feature	base	$+\delta$	$+\delta + \delta\delta$	$+\delta$ LDA	$+\delta + \delta\delta$ LDA
acoustic	MFCC	$\surd$	$\surd$	$\surd$		
	PLP	$\surd$	$\surd$	$\surd$		
articulatory	EMA	$\surd$	$\surd$	$\surd$		
	EMA + EPG + LAR	$\surd$	$\surd$	$\surd$		$\surd$
	net EMA	$\surd$	$\surd$	$\surd$		
combined	MFCC + EMA	$\surd$	$\surd$		$\surd$	$\surd$
	MFCC + net EMA	$\surd$	$\surd$		$\surd$	$\surd$
	PLP + EMA	$\surd$	$\surd$		$\surd$	$\surd$
	PLP + net EMA	$\surd$	$\surd$		$\surd$	$\surd$

Table 5.2: Each tick denotes a feature set which will be used for speaker-dependent classification. All data is from speaker `fsew0` of the MOCHA corpus. Where dimensions exceed 46, equivalent sets are included where linear discriminant analysis (LDA) has been applied for dimensionality reduction.

5.1.2 Some preliminary experiments

Does the small test-set reflect the full corpus?

Figure 5.1 shows a plot of classification results using LDMs with EMA data as features for state dimensions 0 to 20. Accuracies are shown both on the small test set from the original train/test division, and where a 5-fold cross-validation has been employed. It is apparent that accuracies on the 46 utterance test are much more subject to random variation, and are also slightly higher than those where a 5-fold cross-validation has been used. However, the trend for both is that accuracy improves as the state size increases up to a dimension of around 7 where the graphs remain largely static. Figure 5.1 suggests that a more reliable result can be obtained through cross-validation, though a subset of the data is adequate for preliminary experimentation.

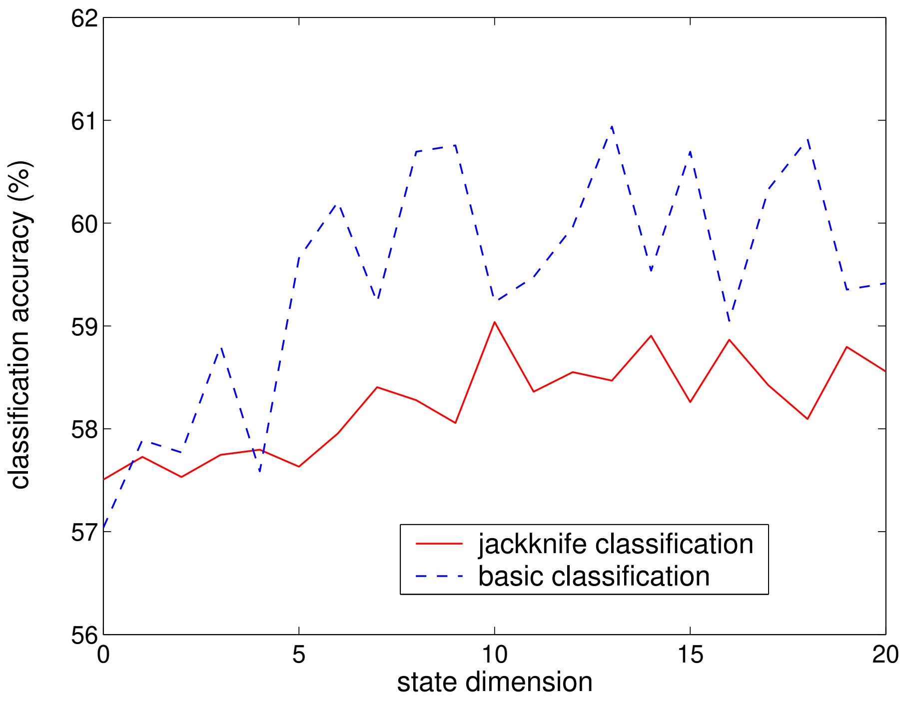

Figure 5.1: Classification accuracies shown for EMA data with LDMs as the acoustic model. The solid red line shows cross-validation classification accuracies for state dimensions of 0 to 20, and the dashed blue line shows classification accuracies on a reduced test-set.

Choosing a frame shift

When preprocessing data for ASR, the experimenter must choose a frame rate (and window size in the case of acoustic data) appropriate to the model being used. A high data-rate means more computation, so a trade-off between speed and accuracy can arise.

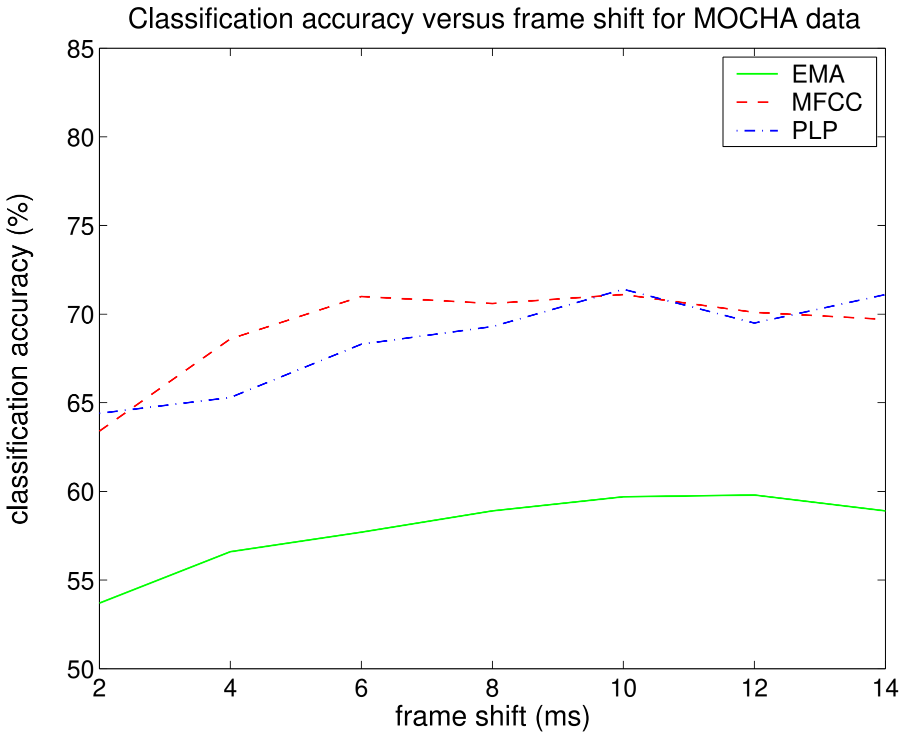

Figure 5.2: Classification accuracies shown for EMA, MFCC and PLP data from the MOCHA corpus for frame shifts between 2 and 14ms. 10ms frames were adopted for all further work

EMA data The EMA data is sampled every 2ms. Down-sampling was performed by first low-pass filtering the data, and then choosing every 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th} or 7^{th} frame to give spacings of between 4ms and 14ms. A set of LDMs with a 9-dimensional state vector was used for classification with no cross-validation and the results are shown in Table 5.3 and Figure 5.2. Larger frame spacings than the original give improved performance. To test if this was due to the smoothing nature of the low-pass filtering,

frame shift(ms)	Classification accuracy		
	EMA	PLP	MFCC
2	53.7%	64.4%	63.4%
4	56.6%	65.3%	68.6%
6	57.7%	68.3%	71.0%
8	58.9%	69.3%	70.6%
10	59.7%	**71.4%**	**71.1%**
12	**59.8%**	69.5%	70.1%
14	58.9%	71.1%	69.7%

Table 5.3: Classification accuracies for systems using LDMs with 9-dimensional states to perform classification on real EMA, MFCC and PLP data from the MOCHA corpus. Frame shifts of 10ms and 12ms give the highest accuracies.

the EMA data was filtered whilst maintaining the original frame spacing. This gives a classification accuracy of 51.3%, lower than with 2ms shift data used raw. Frame-rates of 10ms and 12ms produced the highest classification accuracies.

Acoustic data Section 3.1.2 described the process of producing PLP and MFCC features. Analysis is performed on a series of overlapping windowed regions of the speech signal. A window size of 20ms or 25ms is a common choice for ASR as it provides a reasonable level of smoothing whilst still capturing many of the short-time events. Different systems use differing frame-shifts, for example 16ms (with a 32ms window) in the hybrid ANN/HMM system described in Robinson et al. (2002) or 10ms (with a 25ms window) in a typical HMM system (Young et al. 2002). For this experiment, PLP and MFCC coefficients were generated using 20ms windows on the acoustic signal, and the frame shift varied between 2ms and 14ms.

Classification accuracies for a set of LDMs with a 9-dimensional state are given in Table 5.3 and Table 5.2 alongside those for the EMA data. For both PLP and MFCC features, a 10ms frame-shift gives the highest classification performance. For all features and all further experiments, a 10ms frame-shift will be used.

5.1.3 Experiments using articulatory features

For all experiments which follow, the results of classification based on the single train/test division described in Section 5.1.1 are used to determine the state dimension of LDM and factor analyser models to be used in a 5-way cross-validation. Results using LDMs are shown graphically for each feature set. None of these preliminary results are reported in the text, though are given in full in Appendix E. Where the classification accuracy using LDMs is statistically significantly higher than for both static models, results are shown in bold face.

EMA data Figure 5.3 shows phone classification results using LDMs with EMA, EMA with δs, and EMA with δs and $\delta\delta$s as the features. The state dimension ranges from 0 to 22, where a 0-dimensional state corresponds to a full covariance Gaussian classifier. There is fluctuation in the classification accuracy as the size of the state is varied, though patterns are still apparent. LDM performance for EMA features without δs or $\delta\delta$s improves as the state dimension increases, though remains fairly consistent for models with state sizes of between 5 and 18. The EMA data is a 14 vector, and whilst a much lower dimensioned state offers close to the highest LDM classification performance, it is not until the state dimension is larger than that of the data at around 18 that classification accuracy deteriorates. It is likely that under these conditions, there are more model parameters than can be robustly estimated from the available data. There is not one obvious optimal model dimension, though 12 produces the highest accuracy and so is used as the state size for LDM cross-validation classification. Adding δ and $\delta\delta$ features gives a clear increase in performance. There is still a relationship between state dimension and accuracy, as the accuracy increases with state size up to dimensions of around 16.

The cross-validation classification results in Table 5.4 show that with and without δ features, the modelling of dynamics in an LDM gives an improvement in accuracy compared to static models. The relative error reductions for including dynamic modelling over the best static model are 3.8%, 5.8% and 6.1% for EMA data alone, adding δs and further adding $\delta\delta$s respectively. Surprisingly it is the last of these for which there is the largest relative improvement. The δs and $\delta\delta$s are included for the purposes of adding dynamic information, and so it was expected that in this case there would be the least

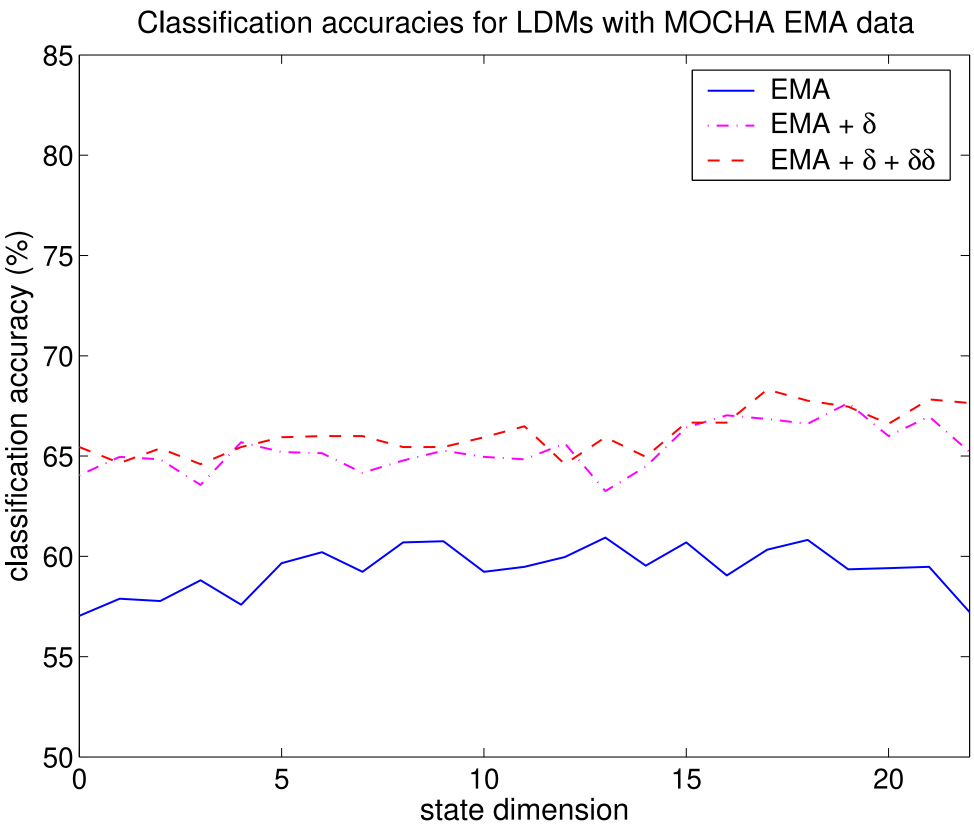

Figure 5.3: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features are EMA data, EMA data with δ coefficients, and EMA data with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		EMA	EMA + δ	EMA + δ + $\delta\delta$
Gaussian		57.3%	63.6%	63.9%
FA	state dim	12	11	16
	accuracy	57.5 %	63.0 %	63.4 %
LDM	state dim	13	19	17
	accuracy	**59.1 %**	**65.7 %**	**66.1 %**

Table 5.4: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are EMA data, EMA data with δ coefficients, and EMA data with δ and $\delta\delta$ coefficients. LDM accuracies in bold face are statistically significantly higher than for either of the static models.

benefit in using a dynamic model. However, LDM models using features which include δs give a higher accuracy than either of the static models which also use $\delta\delta$s.

Extended articulatory features Figure 5.4 shows classification accuracies for LDMs with the state dimension varying from 0 to 24 for the full articulatory features alone, with δs, with δs and $\delta\delta$s, and where LDA has been used for dimensionality reduction on the last of these. For all feature sets, the classification accuracies increase with the dimension of the hidden state, reaching a plateau at around 13. These features consist of the EMA data as used in the previous experiments with the addition of EPG and laryngograph data. These combine to give a 19-dimensional feature vector for which the highest accuracies occur with larger state dimensions than for the EMA data used alone. As in the previous set of experiments, adding δs gives a clear increase in classification accuracy. The effect of further adding $\delta\delta$s or then using LDA for dimensionality reduction is not so apparent from the graph. The trend of improving results as the state dimension increases persists in each case.

For all feature sets, there is a statistically significant increase in the classification accuracy where dynamic models have been used. This shows that improvements are consistent over the test set. In a reversal from the EMA data, the smallest relative error reductions on including dynamic modelling are given where δ parameters are incorporated in the feature set. These are 6.5%, 6.1% and 4.1% for the features used alone, with δs, and with δs and $\delta\delta$s. The smallest gain of 2.4% is found when LDA has been used for post-processing. Given that LDA produces data which is maximally linearly separable, it is not surprising that a static Gaussian is able to give close to the discriminatory power of a dynamic linear Gaussian model.

Figure 5.5 shows a confusion matrix of the classifications made by LDMs with a 19-dimensional state and δs included. This corresponds to the accuracy of 72.1% given in the 2^{nd} column of Table 5.5. The vertical and horizontal axes show true and classified phone identity respectively. The strong diagonal shows that many phones are correctly classified, and the shaded areas off the diagonal display where the errors fall. The strong vertical line in the upper portion of the table shows that vowels and diphthongs are commonly misclassified as schwa, denoted by [`@`]. This is expected, as schwa is one of the most common and highly variable sounds in the English language and can be thought of as the 'neutral position' of the articulators. Another common mistake is to misclassify fricatives and affricates as [`t`]. The parallel diagonal lines about [`b`, `d`, `g`] and [`p`, `t`, `k`] show that whilst the models can distinguish stops according to place and manner of articulation,

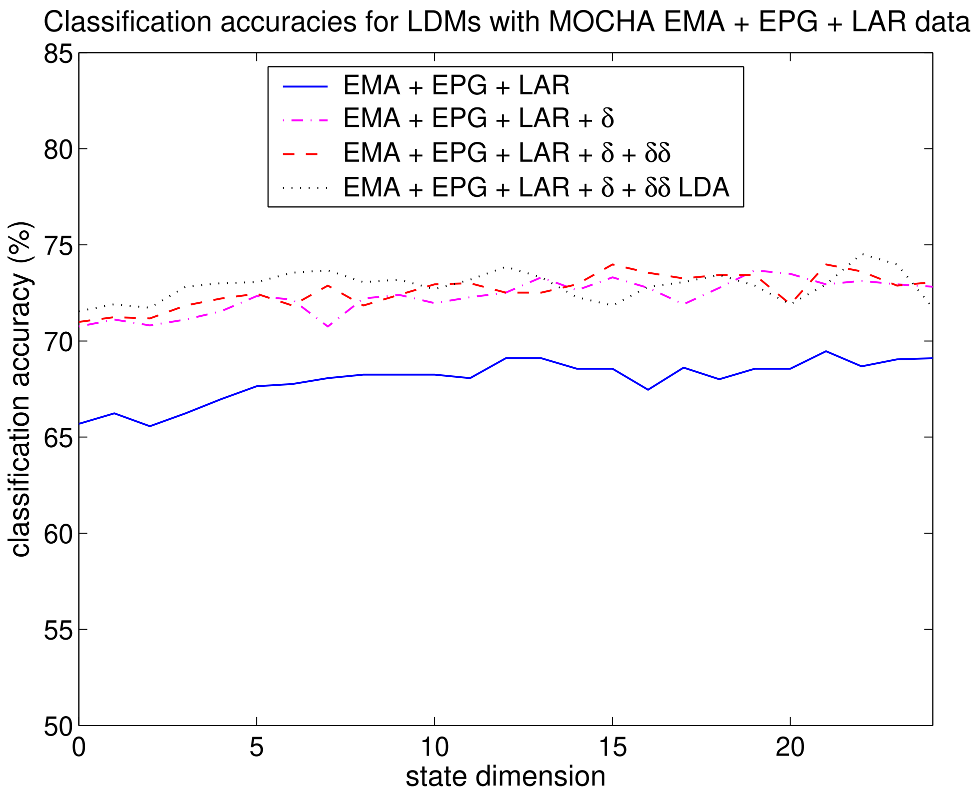

Figure 5.4: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features are the extended articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis) and with the data used raw, or post-processed using LDA.

model and info		artic	$+\delta$	$+\delta+\delta\delta$	$+\delta+\delta\delta$ LDA
Gaussian		66.1 %	70.3 %	70.9 %	71.3 %
FA	state dim	7	21	15	17
	accuracy	65.8 %	70.0 %	70.4 %	71.0 %
LDM	state dim	21	19	15	22
	accuracy	**68.3 %**	**72.1 %**	**72.1 %**	**72.0 %**

Table 5.5: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are the full MOCHA articulatory set comprising EMA, laryngograph and EPG data. Results are shown with the data used raw, or post-processed using LDA. LDM accuracies in bold face are statistically significantly higher than for either of the static models.

making a voiced/voiceless decision is prone to inaccuracy. This would be expected if the features consisted simply of articulatory traces, however the laryngograph data is included to provide voicing information. Other errors due to voicing are also apparent, such as the voiced fricative [`zh`] being frequently classified as its voiceless counterpart [`sh`].

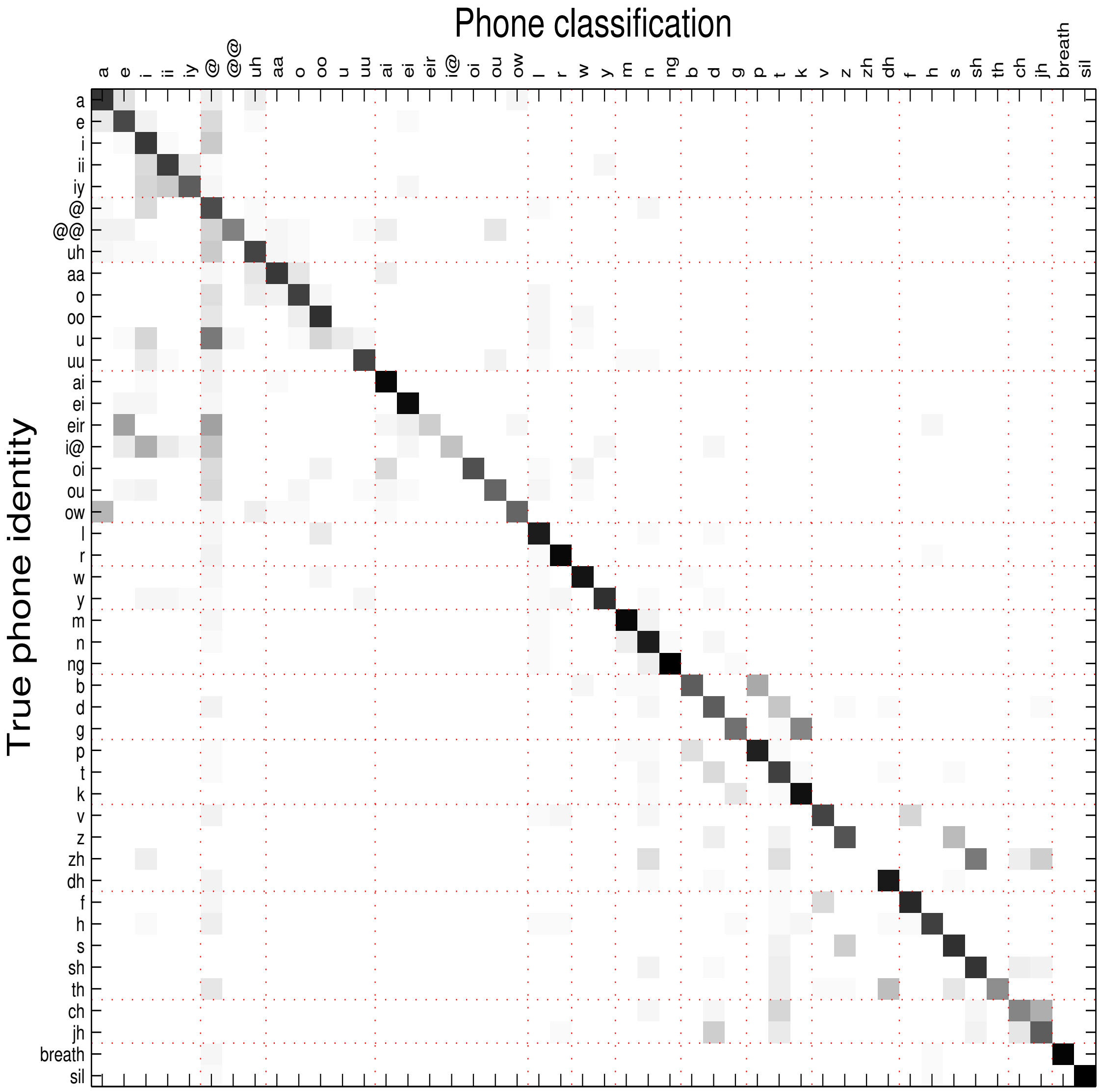

Figure 5.5: This confusion matrix corresponds to the classifications made by LDMs with an articulatory feature set comprising EMA, laryngograph and EPG data with δ parameters and a 19-dimensional state. The overall classification accuracy is 72.1% and is given in Table 5.5. The strong diagonal shows that many phones are correctly classified, though the vertical line in the upper left of the table shows that vowels and diphthongs are commonly misclassified as schwa, denoted by [@]. Another common mistake is to misclassify fricatives and affricates as [t]. The parallel diagonal lines about [b, d, g] and [p, t, k] show that whilst the models can distinguish stops according to place and manner of articulation, making a voiced/voiceless decision is prone to inaccuracy.

Network-recovered EMA data As stated in Section 3.2.1 on page 61, a set of network-recovered EMA data produced using a cross-validation across the 460 sentences was unavailable. Therefore, the results reported here for the network-recovered EMA correspond to the train/test division of the basic classification procedure, as described above on page 116.

The graph in Figure 5.6 shows classification accuracies for LDMs with state dimensions ranging from 0 to 22. These results for the feature sets which include δ and $\delta\delta$ parameters show very little in the way of a trend, though without δs or $\delta\delta$s, an increase in classification accuracy is apparent as the LDM state increases in dimension. Adding δs gives a noticeable improvement in performance, though the effect of further including $\delta\delta$s is not so apparent. Table 5.6 shows the LDM and factor analyser results which give the highest classification accuracy on the validation set, along with those for a Gaussian classifier with a full covariance matrix. In all cases, the dynamic models give highest accuracies, though the improvement over the best static model is not significant where both δ and $\delta\delta$ parameters are included in the feature set. The LDMs give relative error reductions of 4.5%, 2.6% and 0.5% over the best static model when the network-recovered EMA used on its own, with δs and with δs and $\delta\delta$s respectively.

The overall highest accuracy of 59.6% was found for LDMs with 20-dimensional states and δ and $\delta\delta$s included with the features. On the same train/test division, the best performance for the real articulatory data was 67.8%, given by an 18-dimension state LDM. Results using the network-recovered EMA are closer to those based on measured EMA when no δ or $\delta\delta$ parameters are used. These are 59.5% and 57.7% for real and network-recovered features respectively, and use states of dimension 14 and 9.

Figure 5.7 shows a comparison of the classification accuracies found with network-recovered and measured EMA data as features. The errors found using the two feature sets follow similar patterns, though classification of voiced fricatives and stops is considerably worse with network-recovered data. Interestingly, fricatives and stops which do not include voicing give some of the closest accuracies for each feature type. The only instance in which higher accuracies are found using recovered EMA are for the affricates [`ch`] and [`jh`].

Table 5.7 shows the root mean square error (RMSE) for the network predictions of the

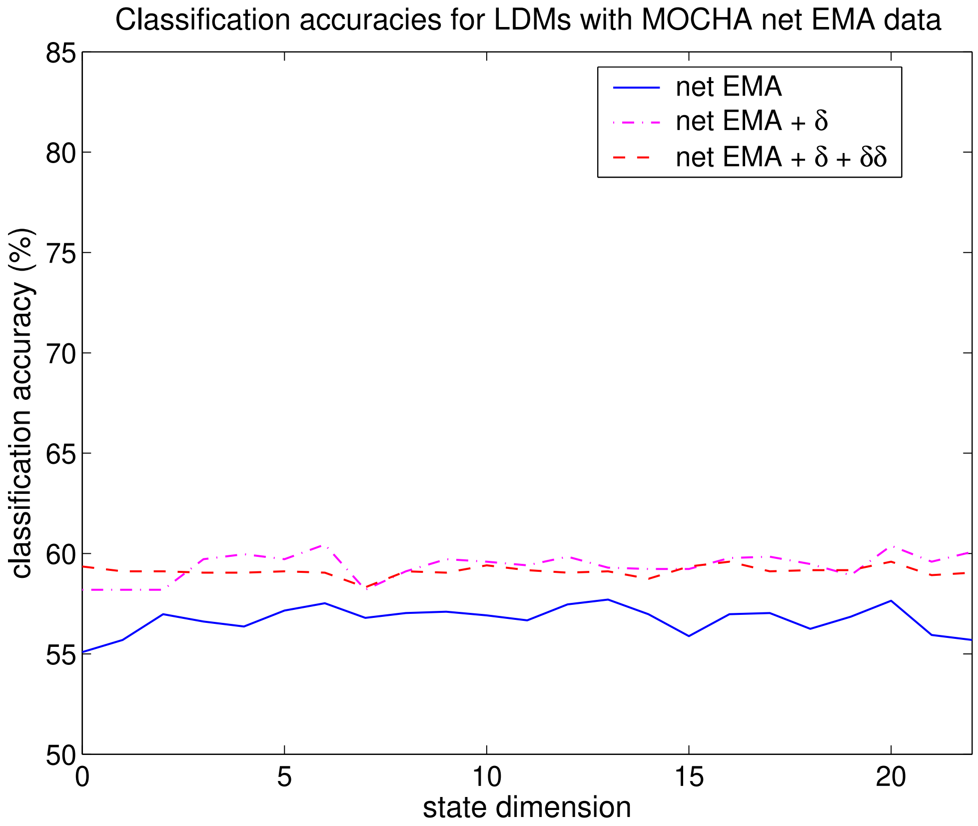

Figure 5.6: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features are network-recovered EMA data used alone, with δ coefficients, and with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		net EMA	net EMA $+\delta$	net EMA $+\delta+\delta\delta$
Gaussian		55.1%	58.2%	59.4%
FA	state dim	16	20	19
	accuracy	49.4%	49.5%	50.5%
LDM	state dim	9	13	20
	accuracy	**57.1%**	**59.3%**	59.6%

Table 5.6: Classification accuracies for systems with LDMs and FA models as the acoustic model. The features are network-recovered EMA data used alone, with δ coefficients, and with δ and $\delta\delta$ coefficients.

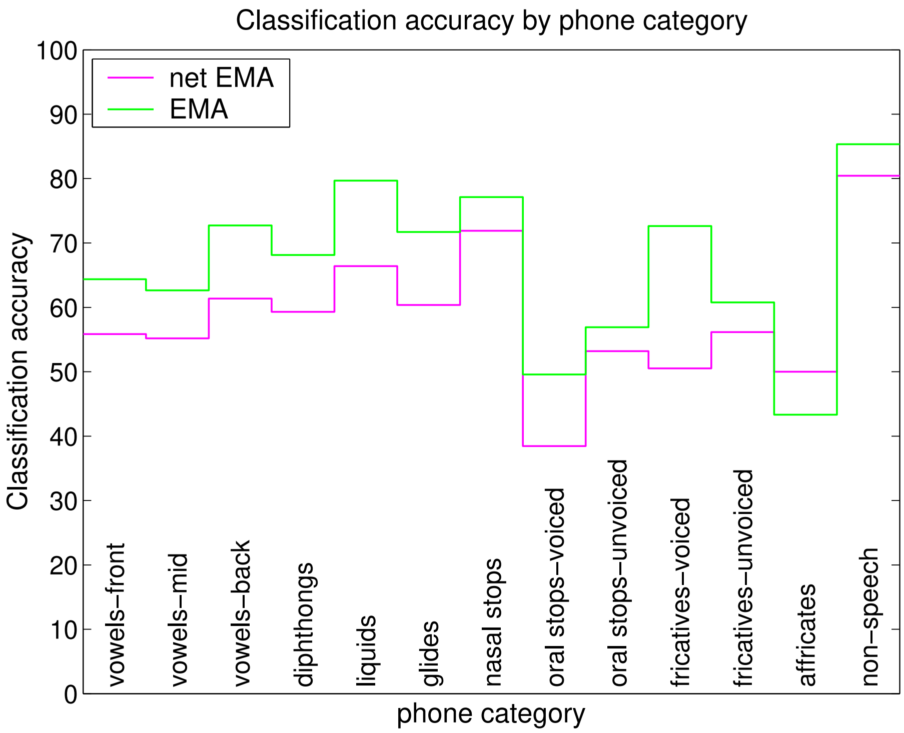

Figure 5.7: Comparison by phone category of the classification accuracies of the highest scoring systems using network-recovered and measured EMA data as features.

EMA data on the test set for each phone category. The networks give the best recovery of data corresponding to affricates and voiceless fricatives, categories which give some of the closest performances under the real and recovered feature sets. Overall, there appears to be some correspondence between accurate recovery of articulation and emulation of the classification performance using real features, though the evidence here is by no means conclusive.

phone category	RMSE
affricates	0.288
fricatives-unvoiced	0.313
diphthongs	0.315
vowels-front	0.317
vowels-mid	0.327
vowels-back	0.334
oral stops-voiced	0.335
fricatives-voiced	0.336
glides	0.341
oral stops-unvoiced	0.348
liquids	0.354
nasal stops	0.363
non-speech	1.022

Table 5.7: Ranked average root mean squared error (RMSE) for the network predictions of the EMA data on the test data for each phone category.

5.1.4 Experiments using acoustic features

PLP features Figure 5.8 shows classification accuracies using LDMs with a PLP parameterization of the acoustics and a state dimension ranging from 0 to 20. Two main observations can be made of this graph. Firstly, there is little performance improvement on adding δ and $\delta\delta$ coefficients. Secondly, it is only PLPs used with no δ or $\delta\delta$ features which show a visible trend of classification accuracy increasing with state dimension.

Table 5.8 shows the cross-validation classification results for LDMs, factor analysers and full covariance Gaussian classifiers. For each of the feature sets, the LDMs provide the highest accuracy, giving relative error decreases of 1.3%, 3.1% and 1.3% over the best static models for PLPs used alone, with δs, and with δs and $\delta\delta$s respectively. However, of these, it is only PLPs with δs which yield a statistically significant performance increase, though with $p < 0.025$ rather than the $p < 0.01$ which is assumed elsewhere. This result of 71.8% is close to the highest classification accuracy found using articulatory features, which was 72.1% for the combined EMA, laryngograph and EPG data with respective δs.

MFCC features The graph in Figure 5.9 shows LDM classification accuracies for MFCC features with and without δ and $\delta\delta$ parameters for state dimensions between 0 and 20. The addition of δs gives a clear increase in performance, though further adding $\delta\delta$s appears to contribute little. When MFCCs are used alone as features, there is a trend of accuracy improving as the state size increases, right up to a dimension of around 10. The increases are more consistent and much less noisy than has been seen for previous feature sets.

The corresponding cross-validation classification results are given in Table 5.9, along with results for full covariance Gaussian classifiers and factor analyser models. With MFCCs used alone and with δ coefficients, the inclusion of dynamic modelling proves useful, giving 3.9% and 4.9% relative error reductions over the highest scoring static model in each case. The highest overall classification accuracy with MFCCs of 75.0% was found when δs are included in the feature set for LDMs with a state dimension of 9. Adding $\delta\delta$ parameters gives a slight decrease in performance, and no evidence to suggest that modelling of dynamics is beneficial in this case. It may be that the resulting 39 dimension feature proves too large for robust estimation of the LDM's parameters.

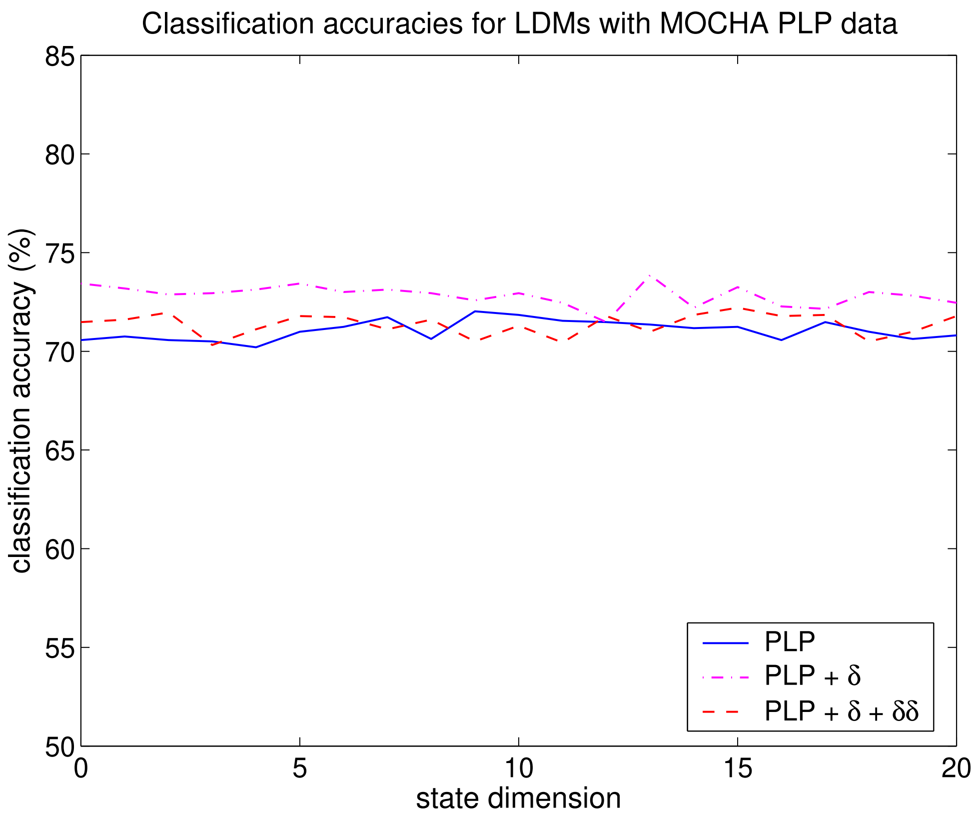

Figure 5.8: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features are PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		PLP	PLP $+\,\delta$	PLP $+\,\delta + \delta\delta$
Gaussian		69.3 %	70.9 %	70.0 %
FA	state dim	17	13	19
	accuracy	68.5 %	70.2%	69.5 %
LDM	state dim	9	13	15
	accuracy	69.7 %	**71.8 %***	70.4 %

Table 5.8: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Result $*$ is significant with $p < 0.025$ rather than with $p < 0.01$ level as assumed elsewhere.

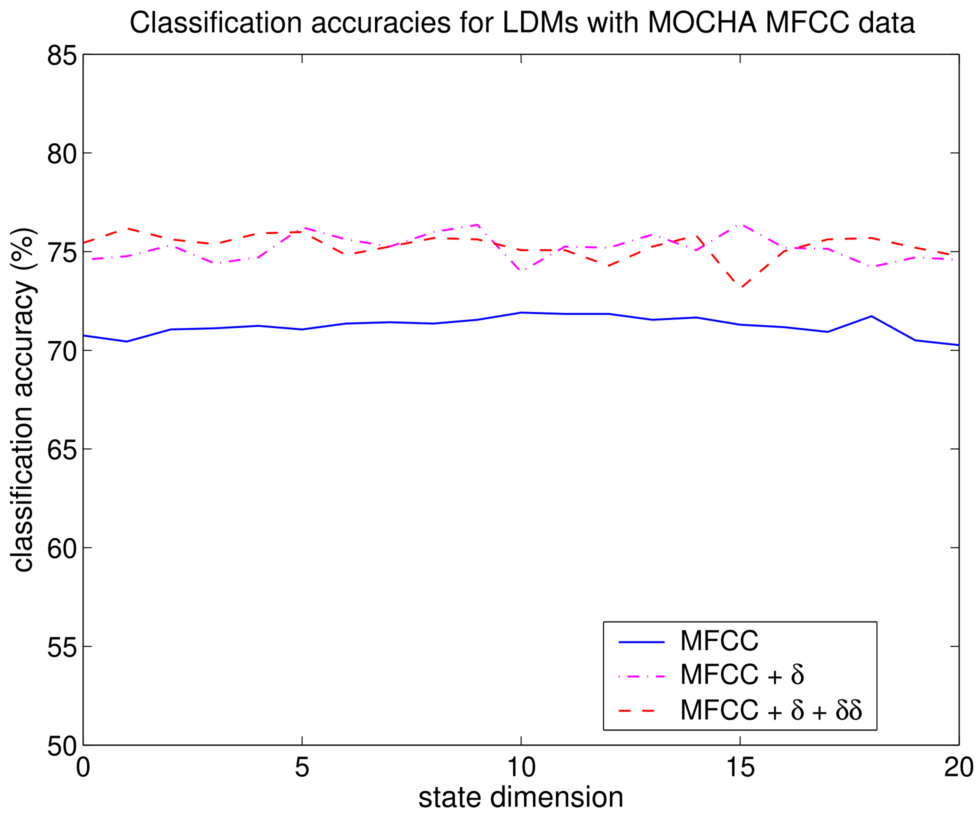

Figure 5.9: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		MFCC	MFCC $+\,\delta$	MFCC $+\,\delta+\delta\delta$
Gaussian		69.3 %	73.7 %	74.3 %
FA	state dim	8	14	14
	accuracy	69.2 %	73.6%	73.8%
LDM	state dim	10	9	1
	accuracy	**70.5 %**	**75.0 %**	74.3%

Table 5.9: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients.

Both of the static models give slight accuracy increases when $\delta\delta$s are added, though still produce lower accuracies than the LDM result which only includes δs.

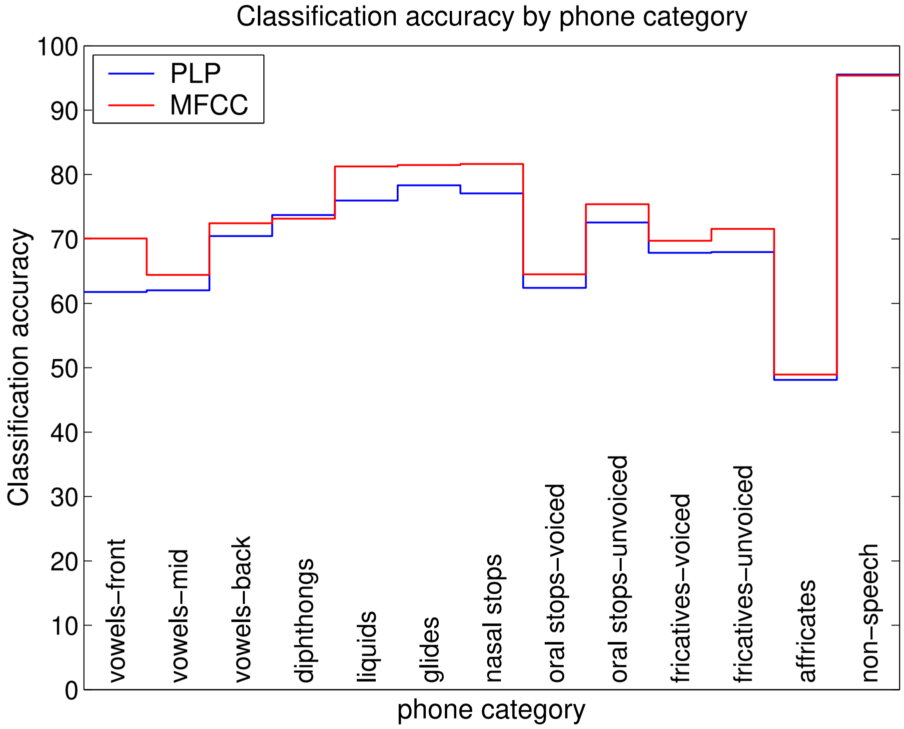

Figure 5.10: Comparison by phone category of the classification accuracies of the highest scoring systems using PLP and MFCC features. The overall accuracies are 71.8% and 75.0% for PLP and MFCC features respectively.

Figure 5.10 gives a pictorial comparison of LDM classification from the best PLP and MFCC systems. In both cases these are where δ parameters are included with the features. The accuracy using MFCCs is higher overall than for PLPs, 75.0% compared to 71.8%, though non-speech and diphthong segments give slightly better classification performance with PLP features. The categories for which there are the largest differences are front vowels and liquids for which the classification accuracies with MFCC features are 8.3% and 5.3% greater than those with PLPs.

Figure 5.11 compares the MFCC + δ system of the previous graph with the most accurate articulatory feature system, in which an LDM with a 19-dimensional state models measured EMA, laryngograph and EPG data. The latter results were originally given in Table 5.5 on page 125. Overall, the acoustic features give a higher classification accu-

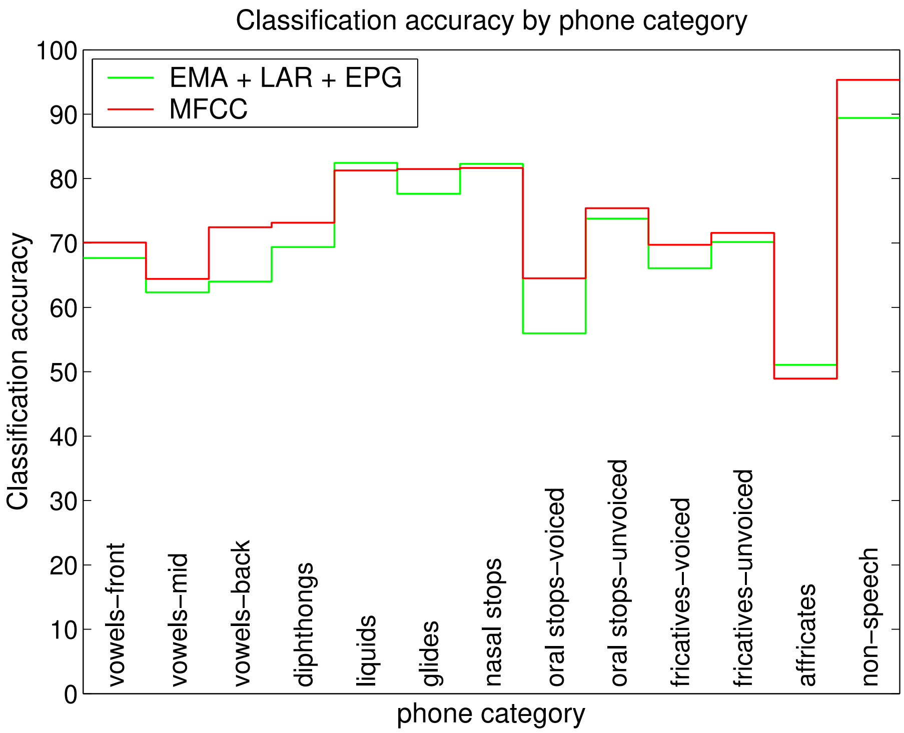

Figure 5.11: Comparison by phone category of the classification accuracies of the highest scoring systems using extended articulatory and MFCC features. The overall accuracies are 72.1% and 75.0% on articulatory and MFCC data respectively.

racy than the articulatory, though the differences are not evenly spread across the phone categories with performance based on the articulatory features being marginally superior for nasal stops, liquids and affricates. The category for which the acoustic features produce the largest improvement over articulatory is voiced oral stops, though the confusion matrix of Figure 5.5 on page 127 suggested that these errors can be attributed to poor voiced/voiceless decisions. Back vowels and non-speech phones are also more accurately classified using acoustic features. The latter category includes silence which it is expected that a model using acoustic features will detect with some certainty.

5.1.5 Experiments combining acoustic data with measured EMA

PLPs and measured EMA data Figure 5.12 shows LDM classification accuracies for feature sets which combine PLP and measured EMA data for state dimensions ranging from 0 to 28. As before, a 0-dimensional state corresponds to a full covariance Gaussian classifier. With an input of plain PLP and EMA features, there is a clear yet noisy increase in classification accuracy as the size of the state vector increases until reaching a dimension of around 15. As before, there is no one optimal dimension, though a state of size 17 produces the best system performance. Adding δ parameters gives a visible improvement in the accuracy of the system, though there is a less marked effect on performance by varying the state dimension. These results appear similar to those found using the sets of features where LDA has been used for dimensionality reduction.

Table 5.10 shows cross-validation classification accuracies for these combinations of PLP and EMA features for LDMs, factor analyser models and full covariance Gaussian classifiers. In all cases, the LDM gives the highest accuracy, though the increases of performance over the best static models are not significant when the data has been post-processed using LDA. The relative error reductions from including a dynamic state for the combined data used alone and with δs are 9.5% and 5.5% respectively. The latter, where features are combined with their δs gives the highest overall accuracy of 79.2%.

MFCCs and measured EMA data Figure 5.13 shows LDM classification accuracies for feature sets which use MFCCs in combination with measured EMA data and state dimension ranging from 0 to 24. The clearest trend is shown where the features do not include δs. The classification accuracy increases with the state size until a dimension of 15 is reached. Performance then tails off when the state attains a dimension of 20. It is likely that in this case, there is insufficient data to produce robust estimates of the parameters of models with larger dimensioned states. Classification accuracy is improved when δs corresponding to both the MFCC and EMA parameters are include in the features, and similar results are found with or without post-processing using LDA.

Table 5.11 shows the cross-validation classification accuracies for LDMs, along with full covariance Gaussian classifiers and factor analyser models. In all cases, LDMs produce the most accurate phone classification, though the increases over the static models are not as

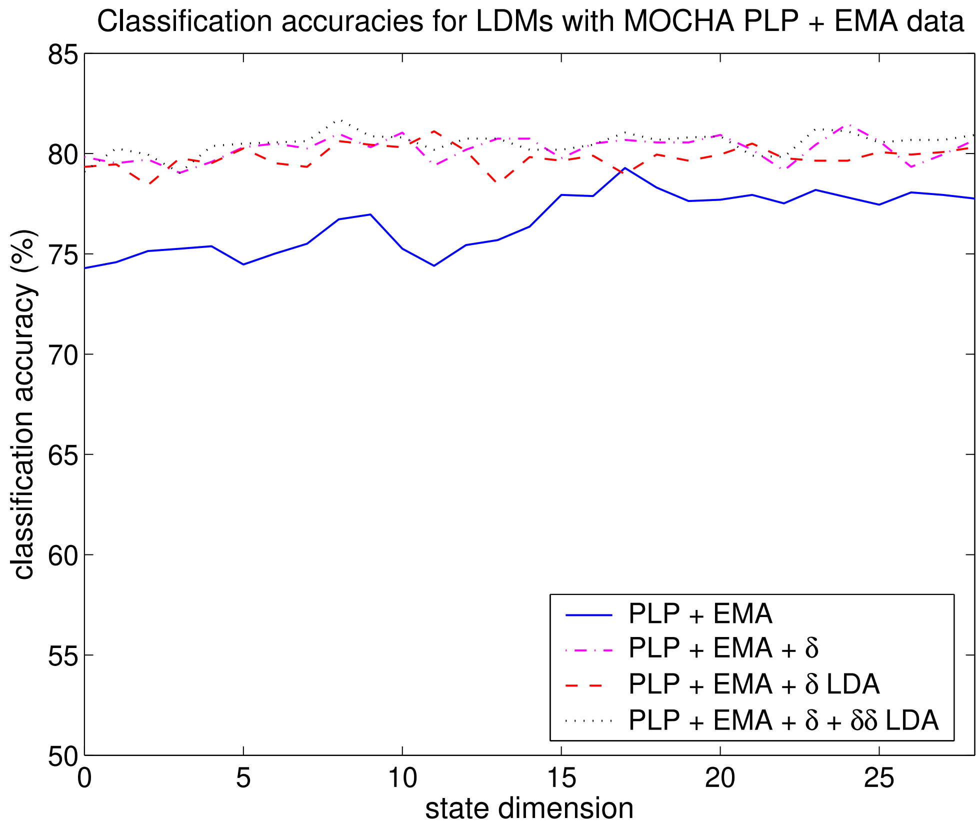

Figure 5.12: Speaker-dependent classification accuracies with LDMs used as the acoustic model. The features are combinations of PLPs and real EMA data, used raw or post-processed using LDA. Classification accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		raw		LDA	
		PLP + EMA	PLP + EMA + δ	PLP + EMA + δ	PLP + EMA+ δ + $\delta\delta$
Gaussian		73.8%	78.0%	78.0%	78.4%
FA	state dim	8	23	12	25
	accuracy	72.7%	77.6%	77.5%	78.2%
LDM	state dim	17	24	11	8
	accuracy	**76.3%**	**79.2%**	78.4%	78.6%

Table 5.10: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are combinations of PLPs and real EMA data, used raw or post-processed using LDA.

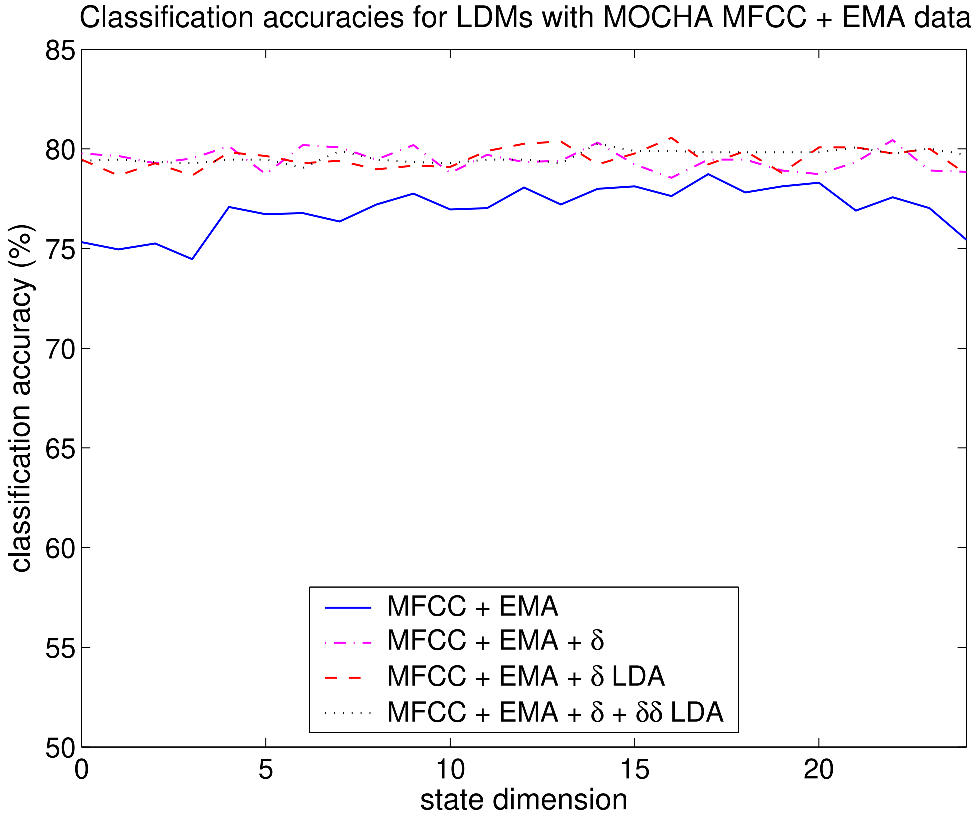

Figure 5.13: Speaker-dependent classification accuracies with LDMs used as the acoustic model. Features are combinations of MFCCs and real EMA, used raw or post-processed using LDA. Classification accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		raw		LDA	
		MFCC + EMA	MFCC + EMA + δ	MFCC + EMA + δ	MFCC + EMA+ δ + $\delta\delta$
Gaussian		73.7%	77.7%	78.1%	77.5%
FA	state dim	13	20	9	12
	accuracy	74.0%	77.6%	77.3%	76.4%
LDM	state dim	17	22	16	14
	accuracy	**75.8%**	**78.4%***	**78.8%***	77.6%

Table 5.11: Cross-validation classification accuracies for systems with LDMs and FA models as the acoustic model. The features are combinations of MFCCs and real EMA data, used raw or post-processed using LDA. Results $*$ are significant with $p < 0.025$ rather than $p < 0.01$.

great as above where PLPs are combined with real EMA data. The gain from including dynamic modelling is largest where no δs are included, with the accuracy increasing from 74.0% to 75.8%, a relative error reduction of 6.9%. Including deltas, with and without LDA dimensionality reduction results in relative error decreases of 3.1% and 3.2% respectively, though these are significant with $p < 0.025$ rather than $p < 0.01$ which is used elsewhere for comparisons. The highest accuracy of 78.8% was obtained with MFCC and EMA features along with all corresponding δs post-processed using LDA as described in Section 3.2.1 on page 62. The models were LDMs with a 16-dimensional state. In this case, applying LDA gives a marginal improvement, as the the equivalent system with same features used raw gives a result of 78.4%.

The effect of adding real articulatory data

features	plain	$+\ \delta$	$+\ \delta$ LDA	$+\ \delta + \delta\delta$
PLP	69.7%	71.8%	–	70.4%
PLP + EMA	**76.3%**	**79.2%**	**78.4%**$^{\Downarrow}$	**78.6%**$^{\Downarrow}$
MFCC	70.5%	75.0%	–	74.3%
MFCC + EMA	**75.8%**	**78.4%**	**78.8%**$^{\Downarrow}$	**76.4%**$^{\Downarrow}$

Table 5.12: This table gives a summary of the classification accuracies found using LDMs with combinations of acoustic and measured articulatory features. The corresponding acoustic-only results are also shown for comparison. $\Downarrow$ denotes that LDA has been used for dimensionality reduction.

Table 5.12 summarises classification accuracies presented in the last two sections: those using just acoustic parameters and those where acoustic and measured articulatory data are combined. These figures show that adding articulatory features improves acoustic-only classification, with the largest error reductions given where the acoustic features are PLPs and LDA is used for post-processing. The relative error reductions in this case are 27.0% and 27.7% where δs and both δs and $\delta\delta$s are included prior to dimensionality reduction. The highest acoustic-only accuracy with PLPs of 71.8% was increased to 79.2%, corresponding to a relative error reduction of 26.2%.

Acoustic-only classification with MFCCs gives higher classification accuracies than

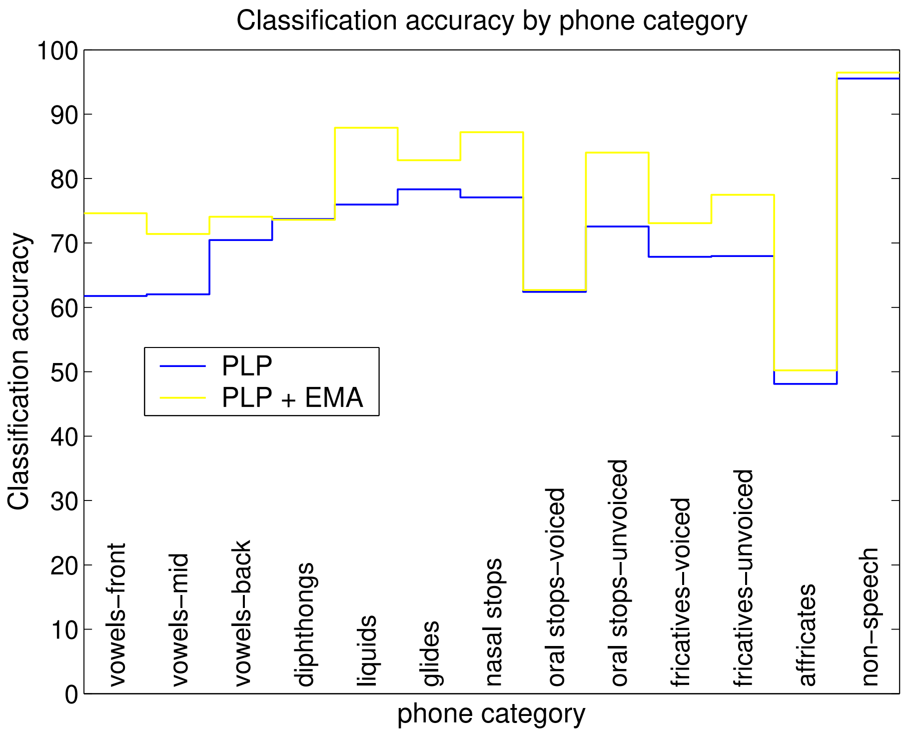

Figure 5.14: Classification accuracies compared by phone category for PLP features with δs and combined PLP and real EMA data again with δs. The latter gives the highest accuracy of the combined feature sets for LDMs with a 24-dimensional state.

with PLPs, and the increases on adding articulatory data are correspondingly lower. The largest improvement was found for PLPs when no δ parameters are included, with a relative error reduction of 18.0%. The highest overall acoustic-only result of 75.0% for MFCCs with δs was increased to 78.8% on addition of real EMA data and δs, and post-processing with linear discriminant analysis. However, the best result for combinations of acoustic and measured articulation of 79.2% is found on PLP and EMA parameters with their respective δs. A breakdown of this result by phone category is shown in Figure 5.14, along with the accuracies obtained using PLPs with δs. Classification performance for diphthongs is slightly higher using only PLPs, and similar for voiced oral stops, affricates and non-speech segments. However, classification of liquids, voiceless stops, voiceless fricatives, nasal stops along with front and mid vowels all give in the region of 10% higher accuracy using the combined features.

Figure 5.15 shows a confusion table corresponding the combined PLP, EMA and δs result. As with the confusion for the extended articulatory feature set on page 127, a common error is to misclassify vowels as schwa. The voiced/voiceless errors are still in evidence, as is misclassification of fricatives and affricates as [t] and [d].

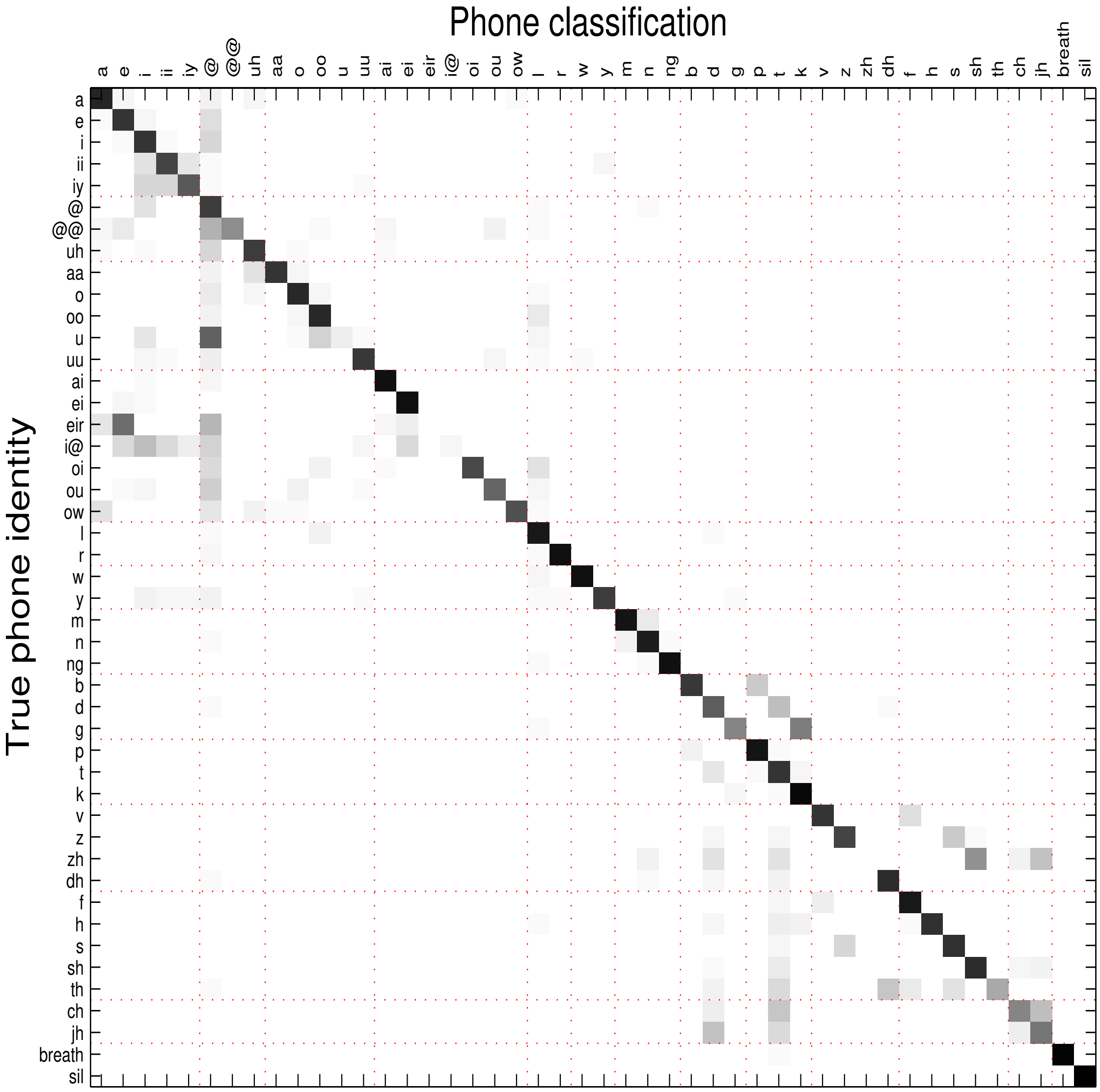

Figure 5.15: This confusion matrix corresponds to the classifications made by LDMs with a combined PLP and real EMA feature set with δ parameters as presented originally in Table 5.10. Common errors include misclassification of vowels as schwa, denoted by [@], and fricatives and affricates as [t] and [d]. Also, voiced/voiceless errors are apparent with voiced oral stops [b, d, g] classified as their voiceless equivalents [p, t, k]

5.1.6 Experiments combining acoustic data with recovered EMA

As with the experiments using network-recovered EMA on page 128, the results in this section will be based on the single train/test division, rather than a 5-fold cross-validation of the data.

PLPs and network-recovered EMA data Figure 5.16 shows classification accuracies for LDMs with combined PLP and recovered EMA feature sets for state dimensions 0 through to 24. The plots show the accuracies for features used alone and with δs. Furthermore, results are given where δ and $\delta\delta$ parameters are included and LDA used for post-processing. The strongest trend for phone discrimination to improve as the state dimension increases is shown where no δs are included in the features. The performance reaches a plateau when the state dimension is in the region of around 12 and then declines again, which may be due to over-parameterization. Including δs and further adding $\delta\delta$s gives a slight improvement to the classification accuracies, though not as much has been seen in other feature sets. Post-processing with LDA does not appear to improve classification accuracy.

Table 5.13 shows the results for which accuracies on the validation data are highest using LDMs and factor analysers, along with results using full covariance Gaussian classifiers. In all cases, the LDMs provide statistically significant improvements in accuracy over the best static model, with the largest relative error reduction of 9.5% being where δs are included and LDA has not been used. It is this combination which also provides the best classification performance of 74.3%.

MFCCs and network-recovered EMA data Figure 5.17 shows an equivalent set of results where MFCCs are combined with network output. Experiments use LDMs with state dimensions ranging from 0 to 24, and features with and without δs, either used raw or subject to LDA post-processing. There is a noticeable trend for the classification accuracies for all features to increase as the state dimension does, up to a size of around 15. Adding δs improves phone discrimination, though applying LDA to the data makes no further contribution.

Table 5.14 shows the best of these results along with corresponding accuracies using

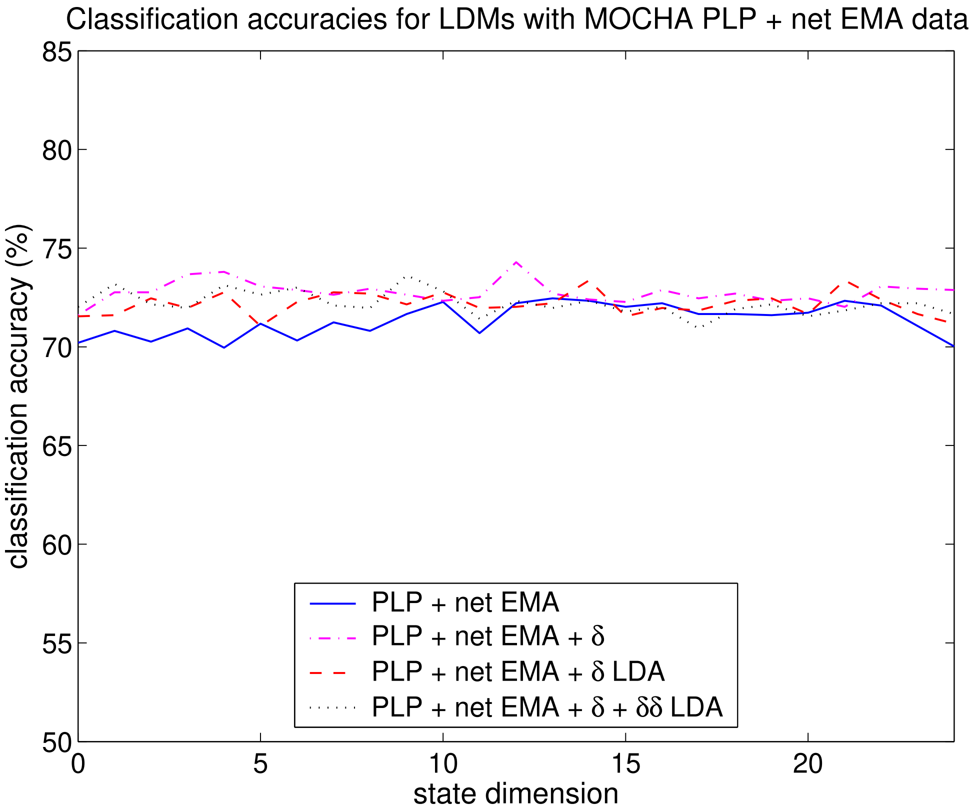

Figure 5.16: Speaker-dependent classification accuracies with LDMs as the acoustic model. The features are combinations of PLPs and network-recovered EMA, used raw or post-processed using LDA. Classification accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		raw		LDA	
		PLP + net EMA	PLP + net EMA + δ	PLP + net EMA + δ	PLP + net EMA+ δ + $\delta\delta$s
Gaussian		70.2%	71.6%	71.5%	72.0%
FA	state dim	20	20	14	18
	accuracy	69.5%	68.2%	70.8%	71.1%
LDM	state dim	20	12	14	18
	accuracy	**71.7%***	**74.3%****	**73.4%***	71.9%

Table 5.13: Classification accuracies for systems using LDMs and FA models. The features are combinations of PLPs and network-recovered EMA data, used raw or post-processed using LDA. Results $*$ and $**$ are significant with $p < 0.025$ and $p < 0.05$ respectively.

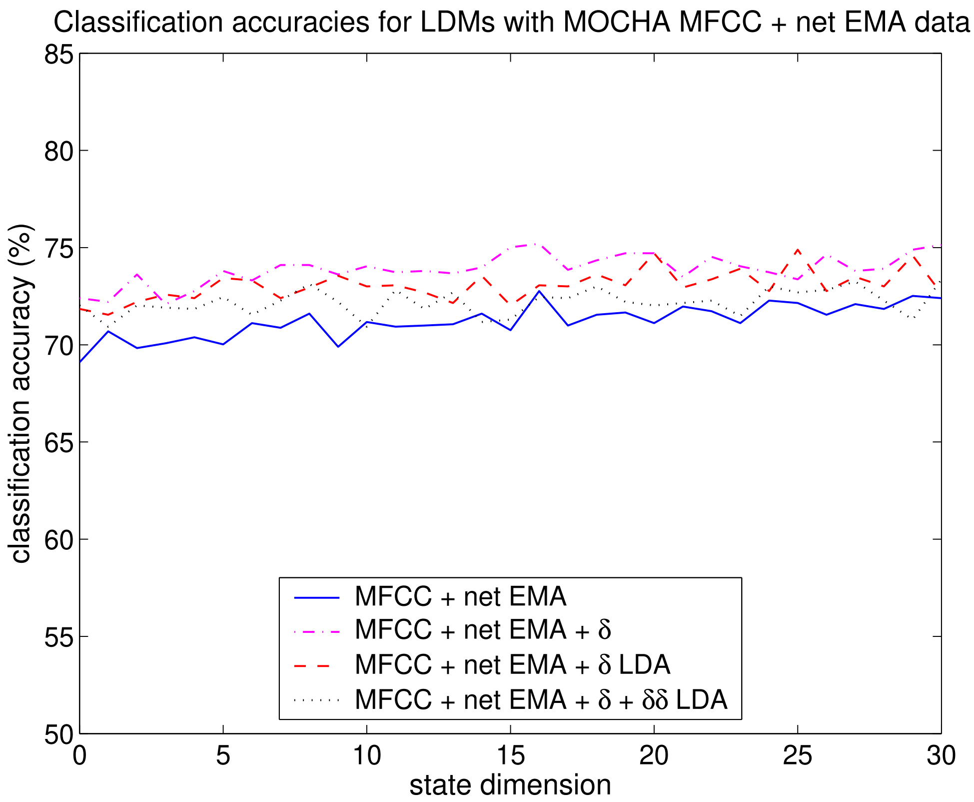

Figure 5.17: Speaker-dependent classification accuracies with LDMs as the acoustic model. The features are combinations of MFCCs and network-recovered EMA, used raw or post-processed using LDA. Classification accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		raw		LDA	
		MFCC + net EMA	MFCC + net EMA + δ	MFCC + net EMA + δ	MFCC + net EMA+ δ + $\delta\delta$s
Gaussian		69.1%	72.4%	71.8%	72.0%
FA	state dim	20	29	19	24
	accuracy	68.7%	71.5%	72.8%	70.7%
LDM	state dim	21	19	16	27
	accuracy	**72.0%***	74.7%	73.1%	**73.2%****

Table 5.14: Classification accuracies for systems using LDMs and FA models. The features are combinations of MFCCs and network-recovered EMA data, used raw or post-processed using LDA. Results $*$ and $**$ are significant with $p < 0.025$ and $p < 0.05$ respectively.

factor analysers and full covariance Gaussian classifiers. The inclusion of a model of the underlying dynamics provides accuracy increases for all feature sets, the largest of these being where LDA has not been used. The relative error reductions for the features used alone and with δs are 10.8% and 10.1% respectively. The latter also gives rise to the most accurate system, where LDMs with a 16-dimensional state correctly classified 75.2% of phones in the test data.

The effect of adding automatically recovered articulatory data

features	plain	+ δ	+ δ LDA	+ δ + $\delta\delta$
PLP	71.4%	72.5%	–	71.3%
PLP + net EMA	71.7%	**74.3%***	73.4%$^\Downarrow$	71.9%$^\Downarrow$
MFCC	70.9%	75.3%	–	**75.6%***
MFCC + net EMA	**72.0%***	74.7%	73.1%$^\Downarrow$	73.2%$^\Downarrow$

Table 5.15: This table gives a summary of the classification accuracies found using LDMs with combinations of acoustic and network-recovered articulatory features. The corresponding acoustic-only results are also shown for comparison. $\Downarrow$ denotes that LDA has been used for dimensionality reduction. Results marked $*$ are significant with $p < 0.05$ rather than $p < 0.01$.

Table 5.15 shows a summary of LDM results using acoustic-only and combined acoustic and automatically recovered articulatory features. Note that all results quoted here correspond to the the train/test division of the basic classification procedure as described on page 116 rather than a 5-fold cross-validation. Following this division, which was used in generating the network-recovered articulatory features, ensures that classification test sentences did not occur in the network training set. Adding network-recovered articulation to each of the PLP feature sets yields increases in classification accuracy. However, it is only the set which includes δs where the increase is statistically significant, and this is with $p < 0.05$ rather than $p < 0.01$. In this case, the highest PLP classification accuracy of 72.5% is increased giving 74.3%, representing a relative error reduction of 6.5%.

Combining MFCC and network-recovered articulation gives an improvement on the acoustic-only baseline where no δ and $\delta\delta$ parameters are included with the features. The classification accuracy is increased from 70.9% to 72.0%, which represents a relative error reduction of 3.8%. A paired t-test shows that the difference between these results is significant with $p < 0.05$. However, the addition of network-recovered articulation gives reductions in classification accuracy where δs or δs and $\delta\delta$s are included with the original MFCC results. For MFCCs with δs and $\delta\delta$s, the reduction is from 75.6% to 73.2%, representing a relative error increase of 3.2%.

These results show that in some cases, adding network-recovered articulatory parameters to acoustic features gives increased classification accuracy. However, none of these give improved performance over the highest acoustic-only accuracy of 75.6%.

Possible drawback of MLP articulatory-inversion for ASR The notion of an articulator being critical or otherwise for the production of a given phone was introduced in Section 3.1.1 on page 51, and Section 4.4.2 on page 108 further discussed the correspondence of criticality of an articulator and its variance. The feed-forward MLP used in the inversion mapping gives an estimate of the conditional average over the articulatory parameters given the acoustic inputs. However, there is no provision to model the natural variation present in the articulation. It may then be that the MLP introduces consistency where there would naturally be none. Variation in the data is a key part of the modelling, as it provides a means for the LDM to estimate the confidence which is placed on each observation stream when likelihoods are calculated. In summary, there may be an overemphasis on data corresponding to non-critical articulators when phonetic identity decisions are made.

An alternative type of network might provide a more useful articulatory inversion mapping for the purposes of recognition. Zacks & Thomas (1994), replaced the sum of squares error function with one which incorporated a measure of correlation. The goal was to encourage capturing articulatory shapes rather than simply mean location. Another approach was taken in Richmond (2001), where mixture density networks were applied to the task of articulatory inversion. Such networks are capable of modelling the one-to-many relationship which is known to exist between acoustic and articulatory parameters. The target variance of a mixture density network with a single mixture component would be able to provide a confidence score on the articulatory data, thereby including in the model the natural variation in measured articulation.

5.1.7 Summary of MOCHA classification experiments

A multitude of classification results were presented in the previous section. The central question which these experiments were designed to address is whether the addition of an explicit model of inter-frame dependency would improve classification performance over that of a static model which assumes framewise independence.

data	model	features	classification accuracy
acoustic-only	static	MFCC + δ + $\delta\delta$	74.3%
	dynamic	MFCC + δ	**75.0%***
acoustic-articulatory	static	PLP + EMA + δ + $\delta\delta$ LDA	78.4%
	dynamic	PLP + EMA + δ	**79.2%***

Table 5.16: Comparison of the overall best results using static and dynamic models on data derived from MOCHA acoustic and acoustic-articulatory data. Result $*$ is significant with $p < 0.025$ rather than with $p < 0.01$ level as assumed elsewhere.

Table 5.16 shows the overall best speaker-dependent classification accuracies using static and dynamic models on acoustic-only and combined acoustic-articulatory data. Note that the systems which include recovered articulation come into the category of using acoustic-only data. In each case, the static and dynamic models give their highest accuracies on different feature vectors. Such a comparison is valid as the original data from which the parameters are generated is identical in each case. Equal care has been taken in optimising the performance of both static and dynamic models, and it is unsurprising that each model type favours certain features. In both cases, the dynamic models give marginally superior performance, significant with $p < 0.025$. Static models give their highest classification accuracies where both δs and $\delta\delta$s are included in the features, where dynamic models use only δs. In this case, the LDM state can be seen to provide and exceed the information encapsulated by $\delta\delta$ parameters.

5.2 Speaker-independent TIMIT classification

Data from the MOCHA corpus offers the unusual opportunity of examining the effect on classification and recognition accuracy of combining acoustic and articulatory data. However, experiments were speaker-dependent and models trained on a little over 15 minutes of speech data. The TIMIT corpus provides over 4 hours of speech data from 630 speakers which has been hand-labelled at the phone level, providing an ideal basis for speaker-independent phone classification and recognition experiments. Well-known benchmark results exist for TIMIT experiments (see Table 2.2 on page 46), allowing meaningful comparisons of results with those of other systems. The training set consists of 124412 phone segments from 462 speakers, compared to the MOCHA `fsew0` training data which comprises 12651 phone segments from a single speaker.

5.2.1 Methodology

The TIMIT corpus has designated training and test sets. However, as in the experiments using MOCHA data, a validation set is required. Of the 462 speakers making up the training data, data from 60 of them was set aside for validation. These are listed in Appendix C, and were chosen with the proportion of speakers from each of the 8 dialect regions following the distribution in the test set. For classification, a number of EM iterations are performed to estimate parameters using the reduced training data and the models stored. Classification accuracy on the validation data is used to determine how many iterations the models should be trained for, and choose a language model scaling factor. Models are then retrained using the combined training and validation data, and data from the test set used to produce a final classification accuracy. Results will be shown pictorially for models with a range of state dimensions, however the final result quoted for each feature set corresponds to the configuration which give the highest classification accuracy on the validation data. All results are given in full in Appendix E.

Lee & Hon (1989) introduced a set of allowable confusions which is commonly used when reporting results on the TIMIT corpus. These are listed in Table 5.17, and provide a means of collapsing the original 61 phone set down to 39. Test set results will be quoted on the 39 phone set, though any validation accuracies will relate to the original 61 phones.

TIMIT allowable confusions
h# pcl tcl kcl bcl dcl gcl q epi pau
m em
ao aa
ax ah ax-h
n en nx
uw ux
ix ih
ng eng
sh zh
l el
hh hv
er axr

Table 5.17: This set of allowable confusions was introduced by Lee & Hon (1989) and is commonly used when quoting results on the TIMIT corpus.

5.2.2 Classification experiments

PLP features Figure 5.18 shows classification accuracies for LDM systems with PLP features used plain, with δs, and with δs and $\delta\delta$s. The state dimension ranges from 0 to 20, where as before, 0 corresponds to a full covariance Gaussian classifier. It is apparent that there is no one optimal size of state vector, though there is a performance improvement as the dimension increases. This is most marked in the case of the PLP features used alone as the size of state vector increases from 1 to 5. With no δs or $\delta\delta$s included, the highest classification accuracy is 67.8%, given by a set of LDMs with a state dimension of 8. Figure 5.19 also shows that adding δ coefficients to the PLP parameters improves phone discrimination, as does further including $\delta\delta$ features. The highest classification accuracies for these feature sets are 71.0% and 72.2%, given by LDMs with 9 and 13-dimensional states respectively.

Table 5.18 gives a summary of the results found with LDMs, factor analysers, and also full covariance Gaussian classifiers. As before, bold face signifies that LDM results are statistically significantly higher than for both of the other models. For all feature sets, the modelling of dynamics offers a modest but consistent improvement in classification accuracy. The largest relative error decrease of LDM against the best static model is

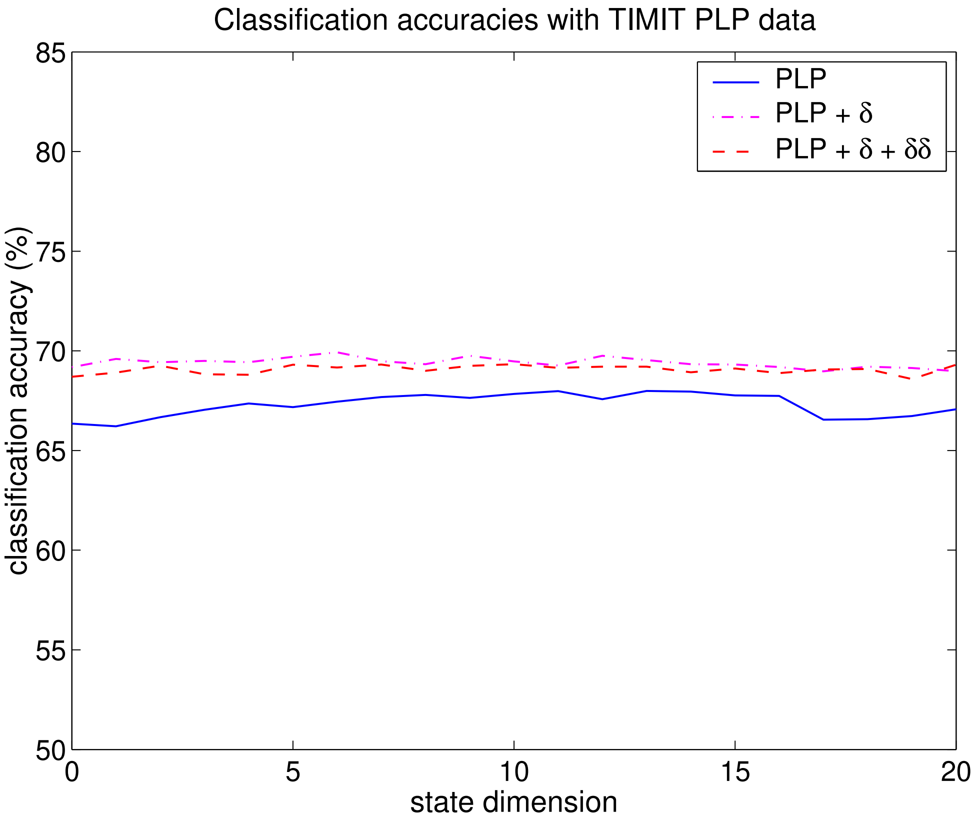

Figure 5.18: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features are PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		PLP	PLP $+\delta$	PLP $+\delta+\delta\delta$
Gaussian		66.3%	70.1%	71.3%
FA	state dim	13	20	20
	accuracy	66.5%	69.3%	70.9%
LDM	state dim	10	9	13
	accuracy	**67.8%**	**71.0%**	**72.2%**

Table 5.18: Classification accuracies for systems with LDMs and FA models as the acoustic model. The features are PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients.

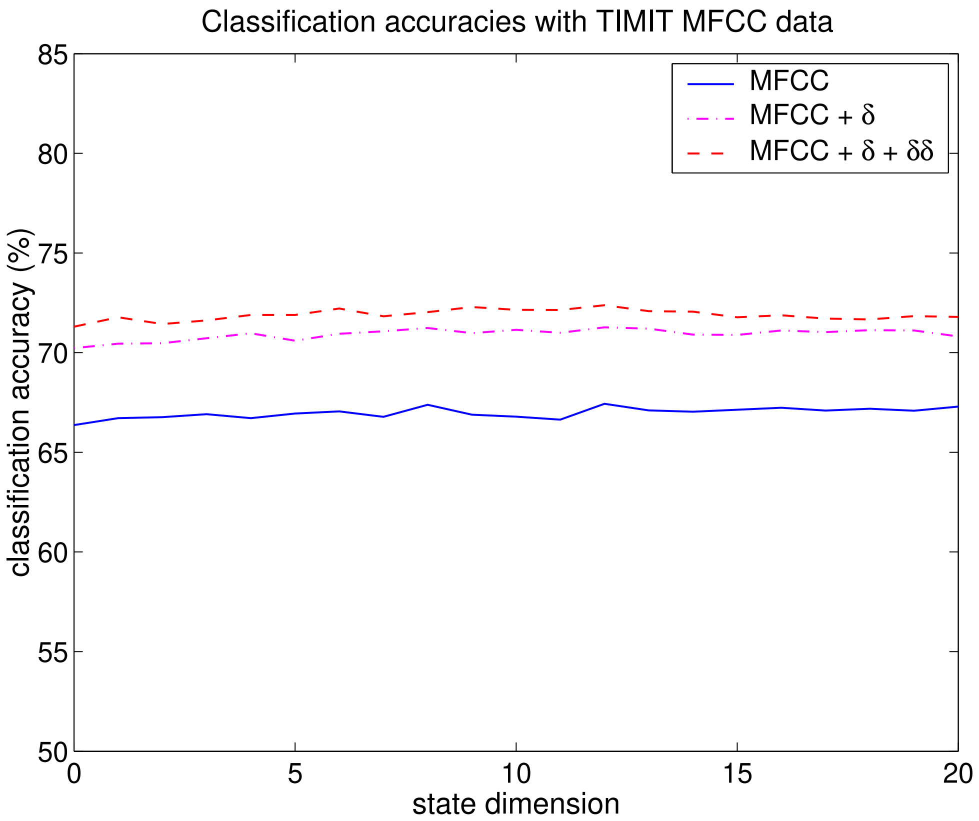

Figure 5.19: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies (y-axis) are shown for a variety of state dimensions (x-axis).

model and info		MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
Gaussian		66.4%	70.2%	71.3%
FA	state dim	10	20	19
	accuracy	66.3%	70.0%	70.7%
LDM	state dim	12	12	9
	accuracy	**67.4%**	**71.3%**	**72.3%**

Table 5.19: Classification accuracies for systems with LDMs and FA models as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. LDM results in bold face denote that the accuracy is statistically significantly higher than for either of the static models.

3.9%, and occurs where no δs are included in the features. This was expected to be so, as the dynamic model should be able to capture some of the same information which the δ parameters are intended to provide. Furthermore, LDMs with δs in the features give similar performance to the static models when δs and $\delta\delta$s are included. The relative error reductions through inclusion of a dynamic state-space are 3.0% and 3.1% when δ and $\delta\delta$ parameters are used in the feature sets.

The graph in Figure 5.18 is noticeably smoother than the equivalent LDM classification results on the MOCHA data, shown in Figure 5.8 on page 133. The TIMIT corpus is considerably larger and the results appear less prone to random fluctuations.

MFCC features Figure 5.19 shows LDM classification of the TIMIT test set using LDMs to model an MFCC parameterization of the acoustics. It is apparent that adding δs improves phone discrimination, as does further including $\delta\delta$s. For each of the three feature sets there is a slight increase in accuracy as the state dimension increases up to around 10. A state vector of size 12 gives the best performance on the validation set for the features used alone and with δs, though 9 provides the highest accuracy where $\delta\delta$s are also included. Table 5.19 compares these results with those for factor analysis models and full covariance Gaussian classifiers. The accuracy increases using LDMs are statistically significant over the best static models on each of the feature sets, giving relative error reductions of 3.0%, 3.7%, and 3.5% with MFCCs used alone, with δs and with both δs and $\delta\delta$s respectively.

The overall highest accuracy of 72.3% for acoustic features was given using a set of LDMs with a 9-dimensional state to characterise MFCCs with δs and $\delta\delta$s. A confusion matrix is given for this result in Figure 5.20 on page 154. Some of the most common errors appear to be misclassifying vowels as [`ix`] and discriminating between [`er`] and [`axr`], which are in fact very similar acoustically. Errors also arise making voiced/voiceless distinctions such as between the fricatives [`zh`] and [`sh`] and also [`z`] and [`s`]. The confusion table in Figure 5.20 also shows that most nasals are classified as [`m`] or [`n`], and the gap in the diagonal shows classification of [`eng`] segments is very poor. However, there are only 4 examples of [`eng`] in the test set.

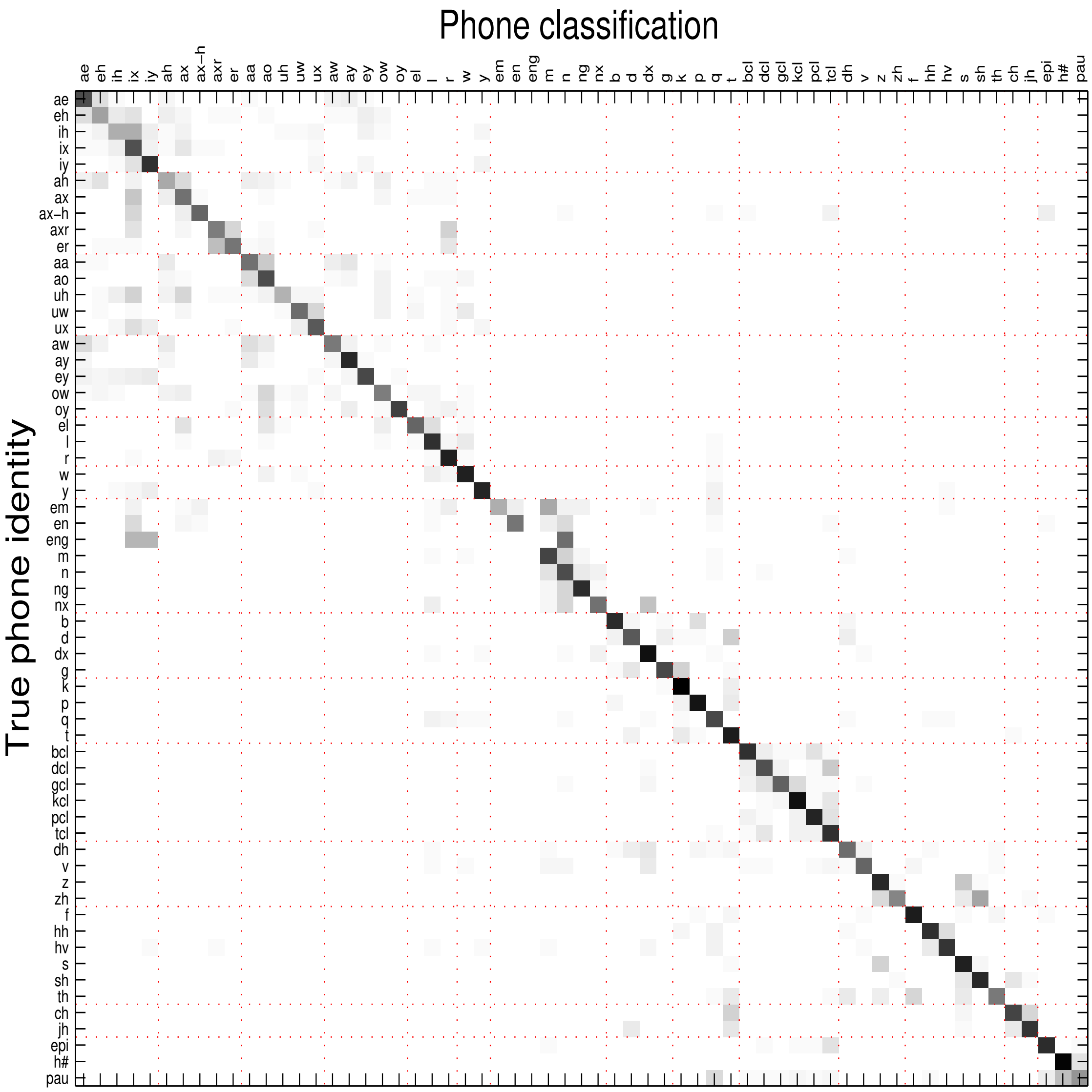

Figure 5.20: This confusion matrix corresponds to the classifications made by LDMs with a 12-dimensional state and MFCC features with both δ and $\delta\delta$ parameters. The accuracy of 72.3% is reported in Table 5.19 on page 152. Some of the most common errors appear to be misclassifying vowels as [`ix`] and discriminating between [`er`] and [`axr`], which are in fact very similar acoustically. Errors also arise making voiced/voiceless distinctions such as between the fricatives [`zh`] and [`sh`] and also [`z`] and [`s`]. Most nasals are classified as [`m`] or [`n`].

Figure 5.21 gives a comparison of the classification accuracy found using PLP and MFCC features, with results broken down by the phone categories used in the comparison of linear and non-linear models in Section 3.4. In both cases δ and $\delta\delta$ parameters are included with the features. The overall accuracies are very similar, being 72.2% and 72.3% for PLPs and MFCCs respectively, though Figure 5.21 shows that there is is variation in the distribution of the errors. Using MFCCs, classification of unvoiced fricatives and diphthongs are slightly over 2% more accurate than the equivalent result based on PLP features. The situation is reversed for glides and voiced fricatives, for which classification is almost 2% higher for PLPs than for MFCCs.

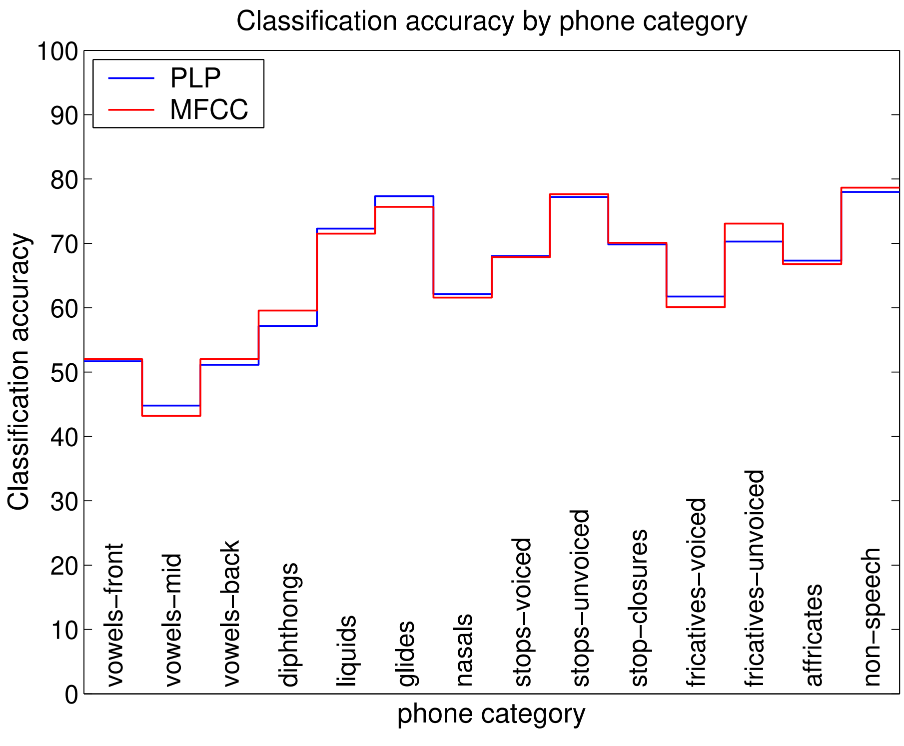

Figure 5.21: The classification accuracies are compared by phone category for PLP and MFCC features. In both cases δs and $\delta\delta$s are included. Across the entire test set the accuracies are 72.2% and 72.3% for PLPs and MFCCs respectively.

5.2.3 Comparing static and dynamic models

Just as in the MOCHA phone classification of Section 5.1, the primary goal of these experiments is to assess the benefit of modelling the correlation between successive frames of speech. Table 5.20 shows the overall highest classification accuracies using static and dynamic models on TIMIT acoustic data. With either MFCC or PLP parameters and corresponding δs and $\delta\delta$s, the highest static model accuracy is 71.3%. LDMs with MFCCs, δs and $\delta\delta$s give a slightly higher, and statistically significant result of 72.3%. This represents a relative error reduction of 3.5%. Unlike the equivalent result on MOCHA data, classification accuracy for LDMs increases on adding $\delta\delta$s to MFCC features with δs. It appears that in this case there is sufficient data to train the extra model parameters.

model	features	classification accuracy
static	MFCC + δ+ $\delta\delta$ / PLP + δ+ $\delta\delta$	71.3%
dynamic	MFCC + δ+ $\delta\delta$	**72.3%**

Table 5.20: Comparison of the overall best results using static and dynamic models on data derived from TIMIT acoustic data.

5.2.4 Checking some assumptions so far

The experiments above have been concerned with examining the impact of adding a dynamic hidden state to static models. The linear dynamic models have been applied in their maximally parameterized form - full observation covariance matrix, sub-space modelling, and no constraint on the form of the state noise. The experiments in this section will compare the various modelling alternatives which were outlined in Section 4.4.3 on page 110. These variants are self-explanatory other than stop 1 and 2 which refer to using full LDMs but ceasing to update the observation noise parameters $\mathbf{v}$ and C after the 1^{st} or 2^{nd} training iterations respectively.

features and dimension		MFCC 13	MFCC $+\delta$ 26	MFCC $+\delta+\delta\delta$ 39
full static Gaussian	state dim	0	0	0
	accuracy	69.3%	73.7%	74.3%
C, D diagonal	state dim	12	14	12
	accuracy	66.5%	71.4%	72.2%
D diagonal	state dim	4	8	11
	accuracy	69.6%	74.3%	**74.7%**
D identity	state dim	3	3	10
	accuracy	70.2%	73.7%	**74.7%**
H identity	state dim	13	26	39
	accuracy	69.7%	74.2%	73.0%
stop 1	state dim	10	9	1
	accuracy	69.8%	73.7%	**74.4%**
stop 2	state dim	10	9	1
	accuracy	70.5%	74.1%	**74.5%**
zero $\mathbf{w}$	state dim	10	9	1
	accuracy	67.0%	74.0%	74.1%
full LDM	state dim	10	9	1
	accuracy	**70.5%**	**75.0%**	74.3%

Table 5.21: Cross-validation classification accuracies for systems with a variety of forms of LDMs as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients derived from the MOCHA corpus. Where the full LDM gives the highest accuracy, results are given in bold face. Otherwise, any variant which gives a better performance than the full LDM is given in bold face

features and dimension		articulatory 19	$+\delta$ 38	$+\delta+\delta\delta$ 57	$+\delta+\delta\delta$ LDA 45
full static Gaussian	state dim	0	0	0	0
	accuracy	66.1%	70.3%	70.9%	71.3
C, D diagonal	state dim	12	19	9	17
	accuracy	64.3%	68.6%	68.5%	70.4%
D diagonal	state dim	15	22	21	12
	accuracy	67.7%	72.1 %	**72.7%**	72.0%
D identity	state dim	16	19	22	18
	accuracy	67.7%	72.0%	72.2%	72.0%
H identity	state dim	19	38	57	45
	accuracy	66.9%	69.6%	66.6%	69.7%
stop 1	state dim	21	19	15	22
	accuracy	66.3%	71.3%	70.3%	70.6%
stop 2	state dim	21	19	15	22
	accuracy	67.2%	71.1%	70.4%	70.7%
zero $\mathbf{w}$	state dim	21	19	15	22
	accuracy	68.2%	71.8%	71.8%	72.0%
full LDM	state dim	21	19	15	22
	accuracy	**68.3%**	**72.1%**	72.1%	**72.0%**

Table 5.22: Cross-validation classification accuracies for systems with a variety of forms of LDMs as the acoustic model. Features are the full MOCHA articulatory set comprising EMA, laryngograph and EPG data. Results are shown with the data used alone, with δs, and with δ and $\delta\delta$s. The latter feature set is either modelled raw, or post-processed using LDA. Where the full LDM gives the highest accuracy, results are given in bold face. Otherwise, any variant which gives a better performance than the full LDM is given in bold face

Tables 5.21 and 5.22 give cross-validation classification results for a number of variations on LDMs with the MOCHA MFCC and extended articulatory feature sets. Where applicable, state dimensions have been chosen on the 46 utterance test set, and then used to obtain full cross-validation results. Similarly, Table 5.23 gives results for the same variations on LDM formulation using TIMIT MFCC features. Where necessary, state dimensions are chosen on the validation data. Full results are given in Appendix E. Table 5.21 shows that on the MOCHA MFCC data, fully specified LDMs give the highest accuracies for MFCCs used alone and with δs. However, when $\delta\delta$s are also included, the

features and dimension		MFCC 13	MFCC $+\delta$ 26	MFCC $+\delta+\delta\delta$ 39
full static Gaussian	state dim	0	0	0
	accuracy	66.4%	70.2%	71.3%
C, D diagonal	state dim	9	11	10
	accuracy	64.6%	68.8%	69.9%
D diagonal	state dim	8	17	17
	accuracy	67.2%	**71.5%**	72.2%
D identity	state dim	19	20	19
	accuracy	67.0%	71.0%	71.9%
H identity	state dim	13	26	39
	accuracy	66.7%	70.6%	71.7%
stop 1	state dim	12	12	9
	accuracy	66.6%	69.9%	71.4%
stop 2	state dim	12	12	9
	accuracy	**67.5%**	70.4%	71.6%
zero $\mathbf{w}$	state dim	12	12	9
	accuracy	66.3%	70.5%	71.8%
full LDM	state dim	12	12	9
	accuracy	67.4%	71.3%	**72.3%**

Table 5.23: Classification accuracies for systems with a variety of forms of LDMs as the acoustic model. The features are MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients derived from the TIMIT corpus. Where the full LDM gives the highest accuracy, results are given in bold face. Otherwise, any variant which gives a better performance than the full LDM is given in bold face

models which use diagonal and identity matrices for the state covariance both give accuracies of 74.7%, higher than the 74.3% found using LDMs with full covariances. Given that the state dimension in the latter case is 1, it may be that MOCHA provides insufficient data to estimate the extra off-diagonal parameters. The equivalent result on the TIMIT corpus, given in Table 5.23, shows that in this case, where there is considerably more data, LDMs with fully specified state covariances give the overall highest accuracy.

Many of the results here are somewhat inconclusive: there is no strong evidence to suggest deviation from a fully parameterized LDM where there is sufficient training data. However, setting diagonal or identity state covariances gives similar accuracies and offers a small degree of computational saving. One result which is worth mentioning is that the inclusion of subspace modelling improves classification accuracy. Using TIMIT MFCC data with δ and $\delta\delta$s, setting $H = I$, gives an LDM classification accuracy of 71.7%, which is statistically significantly lower than that using fully-specified LDMs.

Static models with difference observations compared to LDMs

Section 4.3 on page 102 noted the equivalence between an autoregressive (AR) process and a static model with δ coefficients (Williams 2003), and went on to demonstrate how the modelling provided by an LDM is distinct from these models. The state process in an LDM is first order, though it was shown that the addition of observation noise means that the LDM is not simply a first order model.

The experiment reported below compares classification accuracy on the TIMIT corpus for LDMs with acoustic parameters to that found for static models where simple differences are appended to the feature vectors. The latter provides a model which is analogous to a first-order autoregressive process. Factor analysers are used as the static models.

model	Classification accuracy	
	PLP	MFCC
static model, differences	**68.8%**	**68.6%**
LDM	67.8%	67.4%

Table 5.24: TIMIT classification accuracy for LDMs and static models for which simple differences are included in the observations.

The classification results given in table 5.24 show that for both PLPs and MFCCs, classification accuracy is higher under a static model where simple differences are appended to the features, than for an LDM with only the original features as input. Paired t-tests show that these differences are statistically significant.

It is apparent that in this case, the model of speech signal dynamics given by the LDM does not provide the discriminatory power found by inclusion of differences in the features for a static model. However, in a number of the experiments described above, such as classification of TIMIT MFCCs, LDMs with δs are found to give equal or higher accuracies than the best static model which includes δs and $\delta\delta$s in the features. In these cases, the addition of a dynamic state appears to provide or exceed the extra information contained in the $\delta\delta$ coefficients.

5.3 Variations on the implementation

The experiments above make a straightforward application of LDMs to speech data, with a single model representing the variable-length segments associated with each phone class. Such an approach takes no account of the systematic variation due to segmental duration, which might be dealt with either by constructing models for segments of different lengths, or using some form of duration normalisation. The latter corresponds to the *trajectory invariance* formulation of LDMs used by Digalakis (1992), described in Section 2.3.6 on page 42. Experiments in which duration-normalised fixed-length segments are modelled will be described below in Section 5.3.1.

The manner in which LDMs have been applied thus far in this chapter also assumes that the inter-frame correlations are fixed for the duration of complete segments. There may be an advantage in splitting segments into multiple regimes, each of which is modelled by a separate LDM. This corresponds to the *correlation invariance* assumption which is also described on page 42. Experiments which follow this route will be described in Section 5.3.2 on page 164. Other experiments in this section include combining static and dynamic models in Section 5.3.3 on page 168 and adding an explicit model of phone duration, which is explored in Section 5.3.4 on page 173.

5.3.1 LDMs of fixed-length segments

Making models of fixed-length segments first requires a means of duration normalisation. Possible methods include that of Digalakis (1992), who used a standard linear interpolation, or work by Zavaliagkos et al. (1994), described in Section 2.3.5 on page 38, which employed a truncated discrete cosine transformation. Goldenthal (1994) compared a few methods of segmental duration normalisation, and settled on a fractional linear interpolation which preserves the mapping of endpoints to endpoints and linearly distributes the interior points. A similar method is used in this work.

Of the standard TIMIT LDM-of-phone experiments where δ or $\delta\delta$ coefficients are not included, PLP features give the highest accuracy and so are chosen for this experiment. The result of 67.8% with a 10-dimensional state is used as a baseline. In producing fixed-length segments, the number of frames which each phone token is mapped to can be model-dependent, or fixed for all phone models. Table 5.25 shows TIMIT classification accuracies when an equal number of frames, between 2 and 15, are used in the fixed-length representation of all phone types. Mapping the observations corresponding to each phone segment onto 5 frames gives the highest classification accuracy of 66.7%, though does not reach the baseline.

fixed-length	classification accuracy
2	63.4%
3	65.8%
4	66.4%
5	**66.7%**
6	66.5%
7	66.2%
8	65.7%
9	65.5%
10	65.5%
11	64.8%
12	65.5%
13	65.4%
14	65.3%
15	65.5%
baseline	**67.8%**

Table 5.25: TIMIT classification accuracies in which equal numbers of frames are used in the fixed-length representations of each phone class, and features are PLPs. Results are given for fixed segment lengths of between 2 and 15. Shown in bold face is the highest accuracy gained using fixed-length segments, and the baseline result.

Segment duration varies considerably according to the phone class. For example, the stop release [b] is on average just under 2 frames long whilst segments corresponding to the diphthong [oy] are on average a little under 17 frames. To minimise the impact of the duration normalisation, it may be advantageous to map each phone segment type to a fixed number of frames which is related to its duration distribution. In the experiment reported below, the mean duration along with the 25^{th}, 33^{rd}, 50^{th}, 66^{th} and 75^{th} duration percentiles were used to determine the number of frames which each segment type is mapped to in duration normalisation. Results are given in Table 5.26, and are considerably worse than using a single fixed segment length across all phones, with the highest classification accuracy being 54.3%.

duration statistic providing phone specific trajectory length	classification accuracy
mean	62.0%
25^{th} percentile	54.0%
33^{rd} percentile	**54.3%**
50^{th} percentile	53.5%
66^{th} percentile	53.0%
75^{th} percentile	52.5%
baseline	**67.8%**

Table 5.26: TIMIT Classification results using phone-dependent duration normalisation and PLP features. The fixed number of frames which each segment type was mapped to was determined by the mean duration or one of the percentiles of the phone's duration distribution. The best classification accuracy using duration normalised segments is shown in bold face, as is the baseline.

None of the results which use fixed-length segments reach the variable-length baseline accuracy. It may be that investigating other approaches to duration normalisation will provide improved results. However, any form of duration normalisation is liable to alter inter-frame correlations, which is undesirable for a model which is intended to characterise these dependencies.

5.3.2 Multiple regime LDMs

As mentioned above, Digalakis (1992) used the term *correlation invariance* to describe the assumption of applying distinct LDMs to number of sub-phone regions. A deterministic mapping, dependent on segment duration, dictated the sequence of sub-models which were used to generate each phone. This formulation will be referred to as *multiple regime* (MR) as the abbreviation of correlation invariance, CI, can be confused with the term *context-independent* which is frequently used in speech recognition. Figure 5.22 shows the LDM-of-phone and multiple regime LDMs used to generate a single segment of speech data.

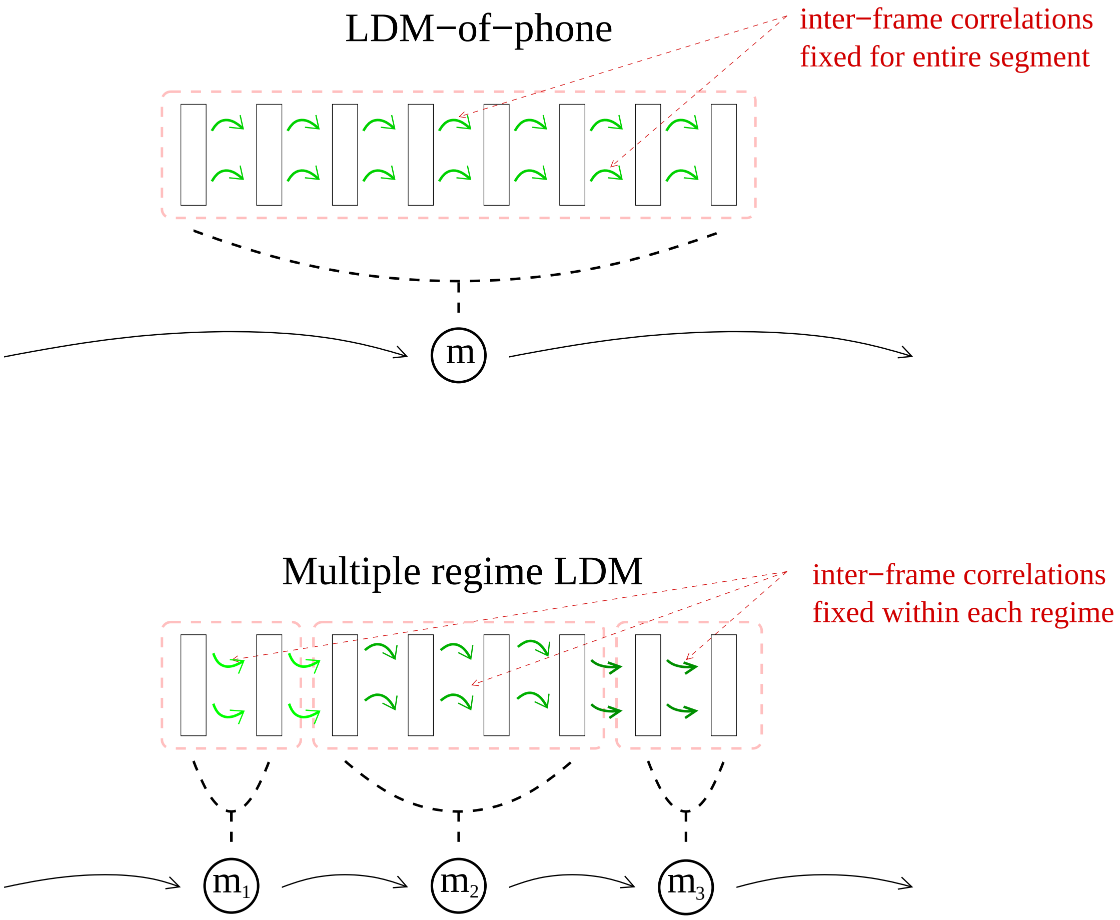

Figure 5.22: In the multiple regime (MR) formulation, segments are split into multiple regimes, each of which is modelled by a separate LDM. The LDM-of-phone approach assumes that inter-frame correlations are fixed across entire segments, whereas in an MR formulation, correlations are static within sub-phone regions. A deterministic mapping dependent on segment length dictates how many frames are spent in each regime.

A multiple regime approach was not taken initially in this work for three reasons:

- using a deterministic, hand-chosen mapping to partition each segment into regions is clearly suboptimal[1]. If such an internal structure is to be used, it should be

[1] This will be discussed in more detail in Section 7.3.3 on page 237

described by some discrete, hidden random variable, and the transition network learnt in a probabilistic manner.

- subdividing segments runs the risk of losing the 'segmental' nature of the modelling. The intention is to model longer sections of speech in which linguistic events occur. Partitioning phone-length segments will produce regions which often consist of only a few frames, and modelling may tend toward the HMM, where models describe short, stationary regions of the parameterized speech signal within which there is little to be gained from an explicit model of dynamics.

- following Occam's Razor which states that 'entities should not be multiplied unnecessarily,' it was considered that the simpler LDM-of-phone model should be investigated first to give a basis for comparison.

Despite reservations over an implementation which uses a deterministic mapping to control the sequence of sub-model regimes, these models do provide an interesting extension to the LDM-of-phone formulation and warrant investigation.

category	regions	division
affricates	2	apportioned equally
fricatives	1	
nasal stops	2	
semivowels and glides	2	
silence	1	
oral stop closures	1	
oral stop releases	3	long tokens mostly final region
vowels	3	apportioned ratio 3:2:3

Table 5.27: Divisions by phone category used in multiple regime LDM experiments.

The mappings used to determine the number of frames which correspond to each sub-model are given in Table 5.27. Models of fricatives, silence and oral stop closures are simply modelled with a single region as the speech signal is considered to be approximately statistically stationary during these sounds. Two regimes corresponding to 'coming in' and 'going out' are used for nasal stops along with semivowels and glides, and for affricates which consist of the combination of a stop and a fricative. Vowels, which are subject to strong contextual variation, are split into 3 regimes modelling 'onset', 'steady state' and 'offset'. Stevens (1999) describes oral stop releases as consisting of 3 distinct regions: a transient, frication at the point of articulation and finally aspiration. Oral stop releases are accordingly split into 3 regimes. All segments are split equally into their chosen number of regions except vowels which are apportioned in the ratio 3:2:3 and the release portions of oral stops in which the 1^{st} and 2^{nd} regions have a maximum duration of 10 and 8 frames respectively. Therefore, in longer oral stop release segments, the largest portion is spent in the final region.

PLP classification accuracy			
model	base	$+\ \delta$	$+\ \delta\ +\ \delta\delta$
LDM-of-phone	67.8%	71.0%	72.2%
full Gaussian MR	68.6%	73.2%	74.2%
state reset MR	68.9%	73.5%	74.4%
state passed MR	**70.2%**	73.6%	74.5%
MFCC classification accuracy			
model	base	$+\ \delta$	$+\ \delta\ +\ \delta\delta$
LDM-of-phone	67.4 %	71.3%	72.3%
full Gaussian MR	68.6 %	73.3%	74.3%
state reset MR	67.9 %	73.3%	74.3%
state passed MR	**69.5 %**	73.7%	74.5%

Table 5.28: Results of using multiple regime LDMs for classification of TIMIT PLP and MFCC features. These results compare standard LDMs with classification in which segments are modelled with multiple regimes. The MR mappings are used for static models, LDMs in which the state is reset between regions, and LDMs in which the state information is passed throughout the length of the segment. Results for which the state-passed LDMs give statistically significant improvements over the other models are shown in bold face.

Table 5.28 shows the results of experiments in which multiple regimes are used within each phone segment for PLPs and MFCCs as features. The baseline classification accuracy found using LDM-of-phone models is shown, along with a result for the case when segments are mapped onto multiple regimes, each of which is modelled by a full covariance Gaussian distribution. This is included as a means of examining whether the dynamic portion of the model still contributes under such a formulation. Multiple region LDM classification results are also presented, both with the state reset between regions (state-reset), and passed throughout the segment (state-passed).

In all cases, static models which use multiple regions outperform standard LDMs of phones. Such a model corresponds to a particular form of HMM in which the state transitions are deterministic given segmental duration, and the output distribution is a unimodal full covariance Gaussian. Accuracy is also improved over the baseline LDM system when multiple regime LDMs are used with the state reset between regions. For all feature sets, the best performance is found with MR LDMs in which state information is passed between sub-phone regions. However, the accuracy increase over all other models is only statistically significant in the absence of δ features.

With MFCC and PLP features respectively and no δs, 2.9% and 4.2% relative error reductions are given using the state-passed MR LDM compared to the next best model. Compared with standard LDMs, the accuracies increase from 67.8% to 69.5% in the case of MFCC features, and 67.8% to 70.2% for PLPs. Including both δs and $\delta\delta$s gives the overall best classification performances, with accuracies of 74.5% for state-passed MR LDMs using each of the acoustic parameterizations. These results are higher than for other models using the same features, though only by a few tenths of a percent. It seems that adding dynamic information in the form of δ and $\delta\delta$ parameters reduces the benefit of including a continuous state process.

5.3.3 Combining static and dynamic models

The final classification accuracies in the last experiment for the multiple region LDM and Gaussian models are similar, however the distribution of errors may differ between static and dynamic models. The following experiments take the results in Table 5.28 for MFCCs with δ and $\delta\delta$ coefficients to examine if combining static and dynamic models will give improved results over using either singly. Figure 5.23 shows the classification accuracies of state-passed multiple region LDMs and multiple region Gaussian models broken down

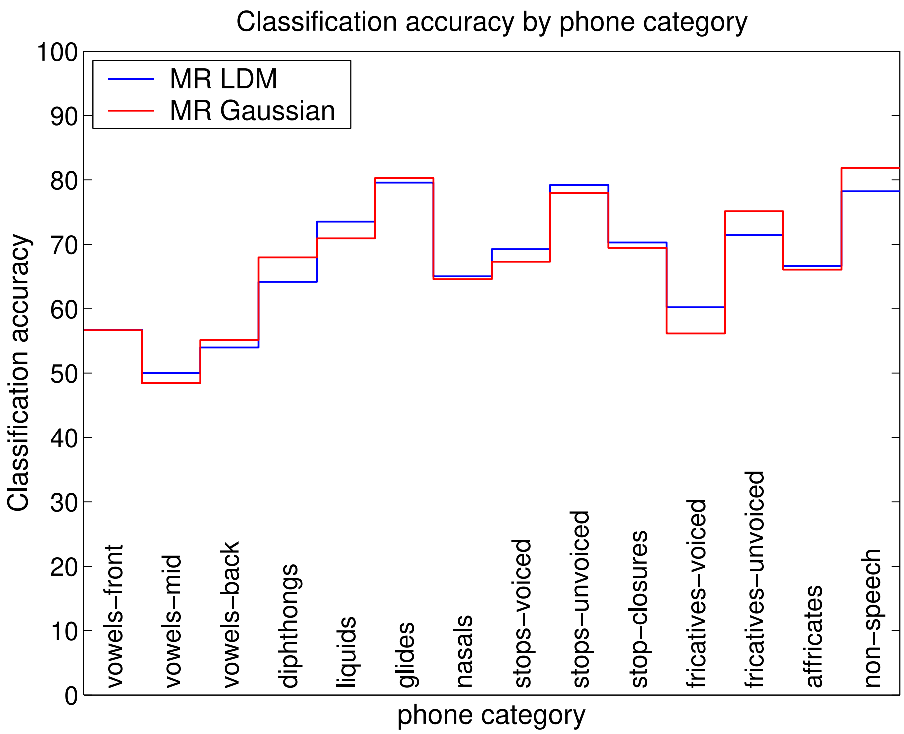

Figure 5.23: The classification accuracies due to state passed MR LDMs and MR Gaussians of Table 5.28 are compared by phone category. Features are MFCCs with δs and $\delta\delta$s.

by phonetic category. The dynamic models give higher classification accuracy for voiced fricatives, liquids and stops, though their static counterparts provide better performance on diphthongs, unvoiced fricatives, and non-speech segments.

Mixed model set

One approach to combining the static and dynamic models is to use a model set composed of a mixture of dynamic and static models. There are a total of 18429 tokens in the validation set, of which 5661 were incorrectly classified by the set of LDM models. Of these errors, 4963 also occurred under the Gaussian models, leaving just over 12% of LDM errors which were distributed differently under the static model. There were 220 tokens which were classified correctly under the static model and incorrectly by the dynamic model. [`h#`] (silence) represented 20% of these, [`s`] 12% and between 5% and 10% by each of [`ae, er, ey, iy`]. Experiments were carried out in which some of the multiple regime

LDMs are replaced by their static models of the same sub-phone regions. In experiment A, dynamic [`h#, s`] are replaced, and in experiment B, [`h#, s, ae, er, ey, iy`] are all replaced with their static model counterparts.

Given that a mismatch between the ranges of the likelihoods produced under different model types is likely, validation accuracies for a number of values of β, a scaling of the static model likelihood are found. These are shown for each experiment in Table 5.29 and Figure 5.24. The highest validation result is 68.2% for a β of 1.0 and dynamic [`h#, s`] replaced with static models. This translates into a 39-phone classification accuracy of 73.9% on the test set. These are significantly lower than the equivalent multiple regime results shown in Table 5.28 on page 167 of 74.3% and 74.5% for static and dynamic models respectively.

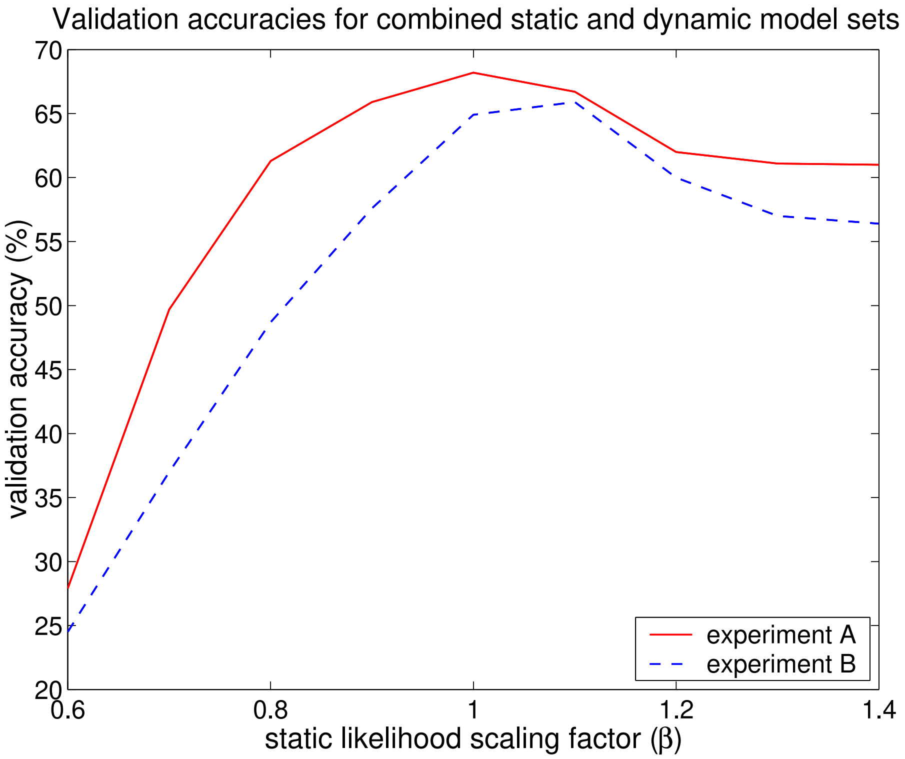

Figure 5.24: Graphs showing the validation accuracies in Table 5.29. Dynamic models are replaced with their static counterparts: [`h#, s`] in A, and [`h#, s, ae, er, ey, iy`] in B. The static model likelihoods are scaled by β

β	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
A	27.9 %	49.7 %	61.3 %	65.9 %	**68.2 %**	66.7 %	62.0 %	61.1 %	61.0 %
B	24.5 %	37.0 %	48.7 %	57.6 %	64.9 %	65.9 %	60.0 %	57.0 %	56.4 %

Table 5.29: 61 phone classification accuracies on the validation set when some dynamic models are replaced with their static counterparts given for a range of values of β, a scaling of the static model likelihood. Dynamic [`h#, s`] are replaced in A and [`h#, s, ae, er, ey, iy`] in B.

Likelihood combination

A number of speech recognition systems have sought to use information from more than one source. For example, Kirchhoff (1998) experimented with a variety of methods of combining the posterior likelihoods from acoustic and articulatory-feature systems for use in recognition. Also, Robinson et al. (2002) used separate neural networks with distinct parameterizations of the acoustic signal as input, the outputs of which were combined to give a scaled acoustic likelihood[2] for use in a hybrid ANN/HMM system. Both of these systems use weighted averages of the log-likelihoods from each source as the final acoustic likelihood. Given that summing log probabilities is equivalent to multiplication of straight probabilities, such an approach makes the simplifying assumption that the classifiers are statistically independent. In the present case, given the similarity between the models which will be combined, this is unlikely to be so. However, there will be no attempt to model interactions for the purposes of this experiment.

Combined likelihoods are computed as $\alpha\, l_s + (1-\alpha) l_d$ where l_s and l_d are the likelihoods under static and dynamic models respectively. Results of classification on the TIMIT validation data are shown in Table 5.30 and Figure 5.25 for $\alpha = 0.0, 0.1, \ldots, 1.0$. A value of $\alpha = 0.3$ gives the highest validation result, and produces a classification accuracy of 74.9%. This result is slightly higher than either the MR Gaussian accuracy of 74.3% and the MR state-passed LDM accuracy of 74.5%, and a paired t-test reveals that in both cases, the increase is statistically significant. Goldenthal (1994), described in the literature review on page 44, reports a context independent TIMIT classification accuracy of 74.2% on the 39 phone set, which is almost identical to the results given here.

[2]See Section 1.3.2 on page 7 for a definition of *scaled likelihood* as used in this context.

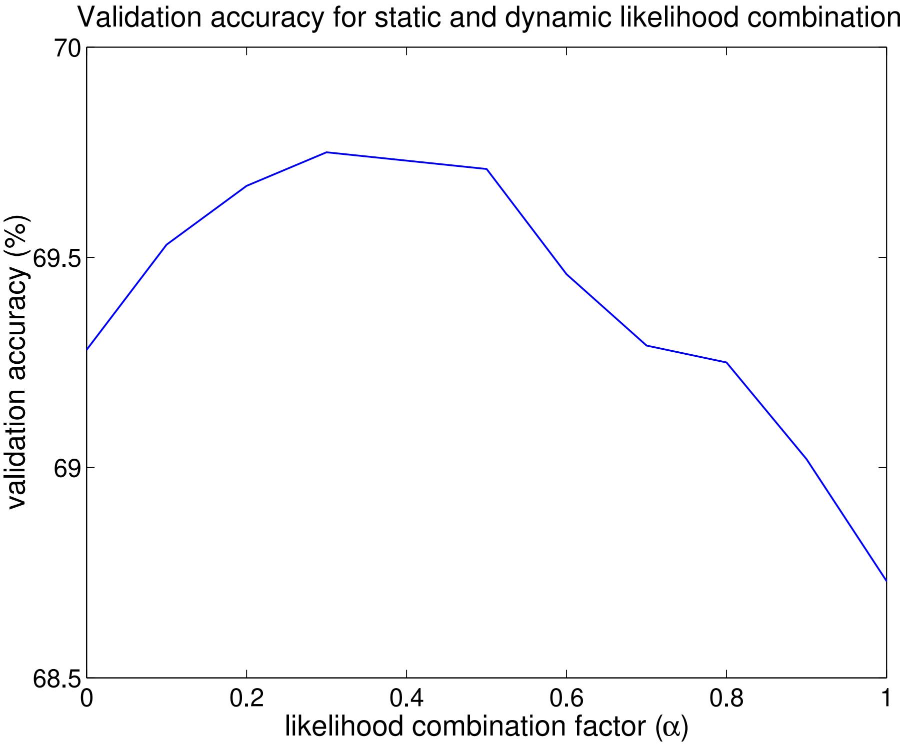

Figure 5.25: This plot shows classification validation accuracies from combining token likelihoods produced under dynamic and static models. New likelihoods are computed as $\alpha\, l_s + (1-\alpha) l_d$ where l_s and l_d are the likelihoods under static and dynamic models respectively. These results shown in this figure correspond to those reported in table 5.30.

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
accuracy	69.3%	69.5%	69.7%	**69.8%**	69.7%	69.7%	69.5%	69.3%	69.3%	69.0%	68.7%

Table 5.30: This table shows classification validation accuracies from combining token likelihoods produced under dynamic and static models. New likelihoods are computed as $\alpha\, l_s + (1-\alpha) l_d$ where l_s and l_d are the likelihoods under static and dynamic models respectively. A value of 0.3 for α gives the highest accuracy. Results are given on the original 61 phone set

5.3.4 Duration model

Experiments thus far have not included an explicit model of phone duration. Such an addition may aid classification accuracy, and two possible models are considered in the experiments which follow. The first uses a Gamma distribution to model the segment durations corresponding to each phone class. The second models the log-durations with a Gaussian distribution.

The parameters describing each candidate duration distribution were estimated from all the segment durations in the TIMIT training set, and experiments to examine the benefits or otherwise of including a model of phone duration follow the classification procedure outlined on page 149. The duration model likelihoods are scaled by κ and added to the acoustic model likelihoods prior to the combination with the language model and Viterbi search.

Single LDM per phone.									
κ	0.0	1.0	2.0	4.0	6.0	7.0	8.0	10.0	12.0
Gamma	66.8%	66.8%	66.9%	66.7%	66.5%	66.3%	66.1%	65.7%	65.2%
log-Gaussian	66.8%	67.0%	67.3%	67.6%	67.7%	**67.7%**	67.7%	67.5%	67.6%
Combined static and dynamic multiple regime models.									
κ	0.0	0.1	0.3	0.6	1.0	2.0	3.0	4.0	10.0
Gamma	69.8%	69.8%	69.9%	69.9%	69.9%	69.7%	69.6%	69.4%	68.2%
log-Gaussian	69.8%	69.8%	69.9%	69.9%	69.9%	70.0%	**70.0%**	70.0%	69.7%

Table 5.31: Classification validation accuracies on adding Gamma and log-Gaussian duration models to both LDM-of-phone and combined static and dynamic multiple region models. The original results which do not include a duration model were given in Table 5.19 on page 152 and Table 5.28 on page 167. Duration model likelihoods are scaled by κ. Note that in both cases these figures correspond to classification accuracies found on the validation set and therefore use the 61 phone set.

Table 5.31 shows 61-phone classification accuracies on the TIMIT validation set on adding log-Gaussian and Gamma distribution duration models to LDM-of-phone and combined static and dynamic multiple region models for a range of values of κ. The original results for each where no duration model is used were given in Table 5.19 on page 152 and Table 5.28 on page 167. The features are MFCCs with δ and $\delta\delta$ parameters. For each model type, the log-Gaussian distribution provides the largest increase in accuracy, though this is marginal in the case of the combined model set.

Table 5.32 shows the results of taking the log-Gaussian duration model with the likelihood scaling found on the validation data and producing classification accuracies on the TIMIT test set. The accuracy for LDM-of-phone models with MFCC, δ and $\delta\delta$ features increases from 72.3% to 73.3%, and the combined static and dynamic models with multiple regions per phone from 74.9% to 75.2%. In both cases, the increases are statistically significant, meaning that the improvement is consistent across the test set. Goldenthal (1994), described in Section 2.3.7 on page 44, compared Gaussian, log-Gaussian and Gamma distributions to model phone segment durations and found that a log-Gaussian model gave the highest likelihood on the TIMIT training set. Context-independent TIMIT classification accuracy was reported to increase from 74.2% to 75.2% on inclusion of a log-Gaussian model of phone duration.

The log-Gaussian duration model will be used in recognition experiments in the following chapter.

models	classification accuracy	
	LDM-of-phone	combined MR models
original	72.3%	74.9%
with durations	**73.3%**	**75.2%**

Table 5.32: Effect on classification performance of including a duration model. In both cases, log-Gaussian models of segmental duration are used. Results in bold face are statistically significantly higher than their equivalents which do not include a duration model.

5.3.5 Summary of TIMIT classification results

The main results of the previous section are summarised in Table 5.33, and correspond to features consisting of MFCCs with δs and $\delta\delta$s. Firstly, classification experiments show that including a dynamic state process gives increased accuracy over an equivalent static model. The 71.3% accuracy obtained using full covariance Gaussian classifiers is statistically significantly lower than the 72.3% found with a set of LDMs. Secondly, subspace modelling, in which a compact representation is made of the data's correlation structure is shown to aid classification. Setting $H = I$, gives an LDM classification accuracy of 71.7%, which again is significantly lower than that using fully-specified LDMs.

With the exception of modelling fixed-length segments, the extensions to the original LDM-of-phone formulation yield increases in classification performance. Adding a log-Gaussian model of phone duration is found to increase classification performance, taking the standard LDM accuracy from 72.3% to 73.3%, a relative error reduction of 3.6%. Using a hand-picked deterministic mapping to govern the transitions between sub-segmental models in the multiple regime formulation is clearly suboptimal. Despite the *ad hoc* nature of this implementation, classification accuracies are higher than those for standard LDMs of phones. However, it is only where no δ or $\delta\delta$ features are included that dynamic models show statistically significant increases over multiple regime static model performance.

model	classification accuracy
static model	71.3%
LDM, no subspace modelling ($H = I$)	71.7%
LDM-of-phone	72.3%
LDM-of-phone + duration	73.3%
MR LDM	74.5%
MR combined LDM and static model	74.9%
MR combined LDM and static model + duration	75.2%

Table 5.33: Summary of the classification results for standard LDMs of phones, multiple regime LDMs and combined LDM and Gaussian multiple regime models. Features are TIMIT MFCCs with δs and $\delta\delta$s.

5.4 Continuous state classification

A state process which is continuous both within and between phone segments would represent a step toward the goal of reflecting the properties of speech production in the chosen acoustic model. Passing state information across model boundaries offers a degree of contextual modelling, and furthermore gives the possibility of modelling longer range dependencies than contained within phone segments.

The spectrograms in Figures 5.26, 5.27 and 5.28 give visual evidence of the potential benefits of a state which is continuous across model boundaries. The first two figures will be familiar from Figures 4.11 and 4.12 on page 107, where spectrograms of the original and LDM-predicted MFCCs corresponding to the sentence ‘Do atypical farmers grow oats?’ were given. The third spectrogram is derived from LDM predictions where the state has been allowed to be continuous across phone boundaries, and as before, the correct model according to the phone labels is used to generate each segment. In all cases, a Mel-warped frequency scale is used, and red corresponds to regions of high energy, whilst blue to low.

A comparison of Figures 5.26 and 5.27 shows that many of spectral characteristics of the acoustic signal are reproduced by the set of LDMs for which the state is reset at the start of each segment. However, spectral transitions are subject to strong boundary effects as each new model takes a few frames to find an appropriate location in state-space. The spectrogram in Figure 5.28 demonstrates how a fully continuous state reduces these effects. For example, the discontinuities in the transition of the first formant through the phones [`ux q ey`] early in the utterance are present where the state is reset, but absent when the state is allowed to be continuous across segment boundaries.

5.4.1 Implementation

As with the multiple regime models above, *state-passed* and *state-reset* will refer to implementations where state statistics are passed across or reset at model boundaries respectively. Training and testing with a fully continuous state require simple modifications of the standard state-reset case as described below. In practice, approximations are made during testing to prevent an exponential increase in the required computation. Alternatively, the state covariances can be reset when boundaries are crossed, but the mean

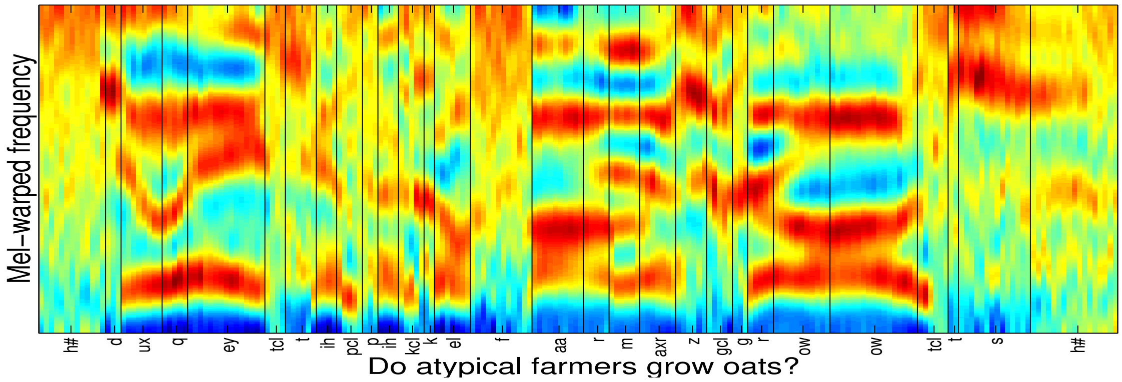

Figure 5.26: A spectrogram generated from the actual MFCCs

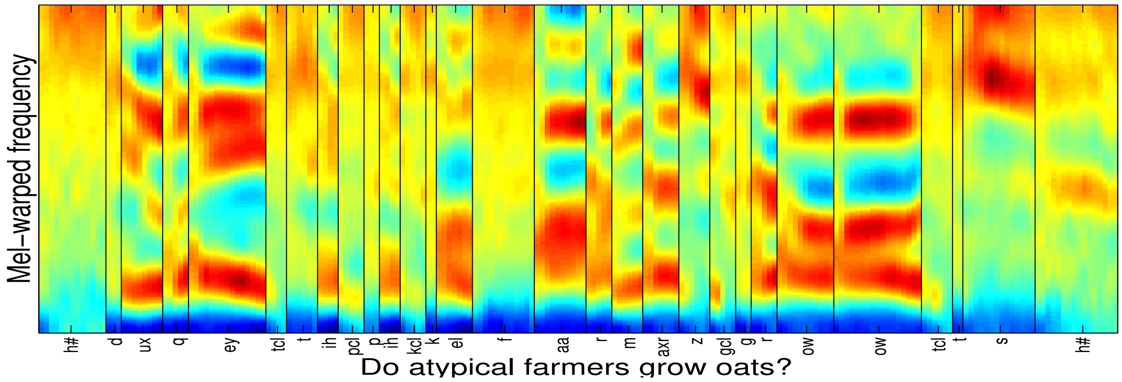

Figure 5.27: A spectrogram generated from predictions made by LDMs during a forward pass through the data.

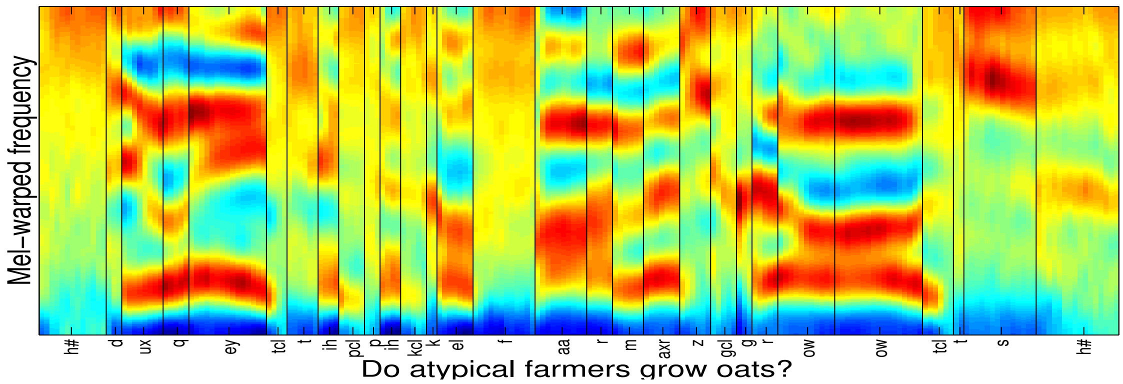

Figure 5.28: A spectrogram generated from the predictions made by LDMs where the state is continuous through the entire utterance.

allowed to be continuous. Some information will still be carried from one phone to the next, but efficient computation can be maintained by pre-computing or caching the 2^{nd} order filter statistics as discussed in Section 4.2.4 on page 99.

Training

The standard EM formulation as described in Section 4.2.2 on page 93 involves computing a set of statistics during the E-step and using these to update the parameters during the M-step. The phone labels are used to extract all the instances of each phone in the training data, and a separate Kalman smoother run over each. An LDM's parameters include the initial state mean $\boldsymbol{\pi}$ and covariance Λ, and it is these which are used to initialise the forward Kalman filtering. The complete-data estimates are accumulated during a backward smoothing pass and used to update parameter estimates in the M-step.

Training whilst allowing the state mean and covariance to be continuous across segment boundaries can be achieved by replacing the forward-backward pass for each token with a forward-backward pass across the entire utterance. In this case, the phone labels dictate which model's parameters are used in each Kalman recursion and then to which model's accumulators the smoothed state statistics are added. The difference from standard training is that at the start of each new segment, the prior on the state $\mathbf{x}_{t|t-1}$ is initialised with predictions based on the final frame of the preceding segment:

$$\mathbf{x}_{t|t-1} \sim N(F\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{w},\ F\Sigma_{t-1|t-1}F^T + D) \tag{5.2}$$

rather than a fixed initial distribution $\mathbf{x}_{t|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$. However, since the initial state remains Gaussian, the linear-Gaussian properties of the LDM are preserved so that estimation and likelihood computations otherwise proceed as normal.

Testing

Passing the state across segment boundaries substantially increases the computation involved in a classification task. With $|\mathcal{M}|$ denoting the number of models between which a classification will be made, and j the number of phone segments in a given utterance, an exhaustive evaluation would require computing the likelihood of $|\mathcal{M}|^j$ possible phone sequences, rather than the $j \times |\mathcal{M}|$ which is required in standard state-reset classification.

Therefore, to keep computational demands within reason, pruning will be needed at some level to discard unlikely partial phone sequences.

The time-asynchronous decoding strategy which will be described in full in Section 6.2 of the following chapter has been adapted to perform state-passed classification. By only allowing paths to finish at the phone-end times as given in the labels, decoding becomes a classification task. Further storing the state mean and covariance as part of any given hypothesis and using these to initialise filter recursions gives continuous state classification. Pruning levels in the decoder can be set to give a minimum of search errors whilst still producing classifications at an acceptable speed[3]. However, using the decoder for this task means that the Viterbi criterion (see Section 6.1.1 on page 187) is applied at phone boundaries. This is not strictly admissible, though is believed to be a reasonable approximation and substantially improves efficiency. This issue is discussed in some detail in Section 7.1.4 on page 227.

5.4.2 Experimentation – MOCHA articulatory data

The first set of experiments are speaker-dependent, use the real EMA data from the MOCHA corpus and correspond to the one-way train/test division as detailed on page 116. LDMs with a state dimension of 14 gave the highest accuracy on the validation set. These results are shown in Table E.2 on page 259 of Appendix E. The corresponding accuracy on the test data was 59.5%, and is used a baseline for the experiments below.

A set of LDMs is trained whilst allowing the state mean to be passed across phone boundaries, after being initialised identically to the models in the baseline experiment. Performing classification with these state-passed models with the standard procedure in which states are reset at the start of each new segment gives an accuracy of 56.2%. A similar experiment, in which both state mean and covariance are passed throughout each utterance in training, but not in testing, gives a classification accuracy of 57.1%. Both of these results are lower than the baseline of 59.5%, which shows that this modified training scheme does not yield performance improvements when using the standard approach to testing.

[3]The TIMIT MFCC continuous state classification results given in Section 5.4.3 were found with the decoder running at around 75 times slower than real-time on a 2.4GHz Pentium P4 processor.

models initialised	classification accuracy	
	mean continuous	mean and cov continuous
from scratch	56.2%	57.1%
1 std iteration	56.9%	57.0%
2 std iterations	58.6%	56.9%
3 std iterations	58.5%	58.9%
4 std iterations	59.8%	60.0%
5 std iterations	56.6%	57.0%

Table 5.34: Classification accuracies for model sets trained with the state mean or both mean and covariances continuous across phone segment boundaries. The standard classification procedure is used where the state is reset at the start of each new segment.

Rather than training state-passed models from scratch, models which have been subject to a few iterations of EM can be used to initialise the state-passed models. Results of such experiments, along with those quoted above are given in Table 5.34. Further training on models after 4 standard EM iterations gives the highest accuracies. These are 59.8% and 60.0% where mean and both mean and covariance are continuous across segment boundaries respectively. These results both provide marginal, though not statistically significant, improvements on the baseline of 59.5%.

The experiments above are subject to a mismatch between training and testing: the state statistics were passed across segment boundaries during parameter estimation, but not during testing. Table 5.35 gives the results of classification using the modified decoder where the state behaviour during testing matches that during training, where either state mean or both mean and covariance are passed between phones. Results in brackets show the corresponding classification accuracies from Table 5.34 where the standard testing procedure is used. The models trained from scratch where both state mean and covariance are carried give equivalent accuracies under both conditions, though in the system where just the state mean is carried, performance deteriorates slightly when training and testing match. Where models had been trained using standard EM, and further trained with a fully continuous state, there is a large reduction in accuracy.

models initialised	classification accuracy	
	mean continuous	mean and cov continuous
from scratch	55.6% (56.2%)	57.0% (57.1%)
4 std iterations	52.0% (59.8%)	53.3% (60.0%)

Table 5.35: Classification accuracies for model sets trained with the state mean or both mean and covariance continuous across phone boundaries. State resetting during testing matches that which was used during training. Accuracies in brackets are equivalent results using the standard classification procedure in which state statistics are reset at the beginning of each new segment.

5.4.3 Experimentation – TIMIT acoustic data

Continuous state classification experiments were also performed on acoustic data from the TIMIT corpus. For MFCCs with no δs or $\delta\delta$s, Table 5.19 on page 152 shows that the highest LDM-of-phone classification accuracy of 67.4% was given using a set of models with a state dimension of 12. The classification results presented in this section are for the TIMIT core test set as described on page 211, and the language models are the backed-off bigrams which will be used in the recognition experiments of Chapter 6. Otherwise, the classification procedure remains as described in Section 5.2.1 on page 149. Standard state-reset classification with these different language models and test set also gave an accuracy of 67.4%, and will provide the baseline for this set of experiments.

The LDMs were initialised to have starting values which were identical to those used when training the models which provide the baseline result. Models were trained from scratch with both state means and covariances passed over segment boundaries, rather than including any iterations of EM where states were reset at the start of segments. The experiments of the previous section showed that this mixed training approach resulted in marginal though not statistically significant increases in accuracy using standard state-reset testing, though when the state was also passed between segments during testing, the lowest classification accuracies were found.

The results are presented in Table 5.36 and show that the highest accuracy of 67.4% is given by the baseline result where the state is reset at the start of each new segment in both training and testing. State-passed training followed by state-reset testing results

implementation training	implementation testing	classification accuracy
state reset	state reset	67.4%
state passed	state reset	66.0%
state passed	state passed	67.0%
state passed	state passed, correct likelihood	66.7%

Table 5.36: Features are TIMIT MFCCs with δs and $\delta\delta$s. Both state mean and cov

in an accuracy of 66.0%, though using these same models and passing the state between segments during testing gives the improved result of 67.0%. The latter is close to the baseline, and shows that in this case, a mismatch between training and testing caused a reduction in performance.

Section 4.2.3 on page 97 showed that a modified likelihood calculation gave higher classification accuracies for shorter phones. With the state continuous across entire utterances, there may be an advantage by re-including the contribution of state covariance in normalising the prediction errors. The last result of Table 5.36 shows that in fact, this causes a slight reduction in classification accuracy, giving 66.7% rather than the 67.0% found using the modified likelihood calculation used in this work.

5.4.4 Summary of continuous state classification experiments

These result of the last two sections show that in the current implementation, classification accuracy is reduced by allowing the state statistics to be continuous across phone boundaries. On MOCHA EMA data, allowing the state to be fully continuous resulted in an accuracy of 57.0%, where models in which the state was reset between segments gave 59.5%. Training models with a single state-passed iteration after initially training with a number of state-reset iterations gave a classification accuracy of 60.0%. This is higher than the baseline of 59.5%, though not statistically significantly so.

On TIMIT acoustic data, the accuracies were similar under state-passed and state-reset implementations of the models, 66.7% and 67.0% respectively. For this data, a mismatch in the state behaviour during training and testing was found to cause a reduction in performance. Section 7.1.4 on page 224 discusses these results further, and suggests ways in which a fully continuous state might prove useful in the future.

Chapter 6

LDMs for recognition of speech

6.1 Decoding for speech recognition

Automatic speech recognition ultimately centres around a search problem: given the probabilities from an acoustic model, pronunciation model, language model and perhaps duration model, the most likely sequence of words must be found. In light of the number of possible word sequences, each of which is subject to a large number of potential time-alignments, an exhaustive search is impractical even for modest sized tasks. Decoding, as this search is known, therefore becomes an exercise in judicious investigation of the search space in which time and memory requirements must be minimised without introducing too many new errors.

There are two main approaches to implementing such a search in ASR. The first is time-synchronous forward dynamic programming, or Viterbi decoding, where all hypotheses are evaluated at a given time before the search proceeds to the next time. The second is time-asynchronous $A*$ search, where regardless of time, the best candidate hypothesis is extended.

With $\mathcal{W} = w_1^j = \{w_1, \ldots, w_j\}$ denoting a word sequence and $\mathcal{Y} = \mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ an observation sequence, Renals & Hochberg (1999) *inter alia* define the task of decoding as finding the maximum a posteriori (MAP) probability of words $\mathcal{W}$ given observations $\mathcal{Y}$:

$$\mathcal{W}^* = \underset{\mathcal{W}}{\text{argmax}} P(\mathcal{W}|\mathcal{Y}) \tag{6.1}$$

However, $P(\mathcal{W}|\mathcal{Y})$ is not a quantity which can generally be computed directly, and so Bayes' rule is used to decompose 6.1 into a product of the acoustic model likelihood and language model probability:

$$P(\mathcal{W}|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{W})P(\mathcal{W})}{p(\mathcal{Y})} \tag{6.2}$$

$$\propto p(\mathcal{Y}|\mathcal{W})P(\mathcal{W}) \tag{6.3}$$

For the purposes of decoding, $p(\mathcal{Y})$ can be omitted as it is independent of the words $\mathcal{W}$. Letting $\mathcal{M} = m_1^k = \{m_1, \ldots, m_k\}$ denote the concatenation of a series of sub-word models which together account for the full observation sequence $\mathcal{Y}$ and produce the word sequence $\mathcal{W}$, 6.3 can further be decomposed as:

$$P(\mathcal{W}|\mathcal{Y}) \propto \sum_{all \mathcal{M}} p(\mathcal{Y}|\mathcal{W},\mathcal{M})P(\mathcal{M}|\mathcal{W})P(\mathcal{W}) \tag{6.4}$$

Given that the sequence of models was chosen to represent the word sequence, $\mathcal{W}$ is implicit in $\mathcal{M}$, and therefore $p(\mathcal{Y}|\mathcal{W},\mathcal{M}) = p(\mathcal{Y}|\mathcal{M})$. This allows the MAP criterion of 6.1 to be re-written as:

$$\mathcal{W}^* = \underset{\mathcal{W}}{\operatorname{argmax}} \left\{ P(\mathcal{W}) \sum_{all \mathcal{M}} p(\mathcal{Y}|\mathcal{M})P(\mathcal{M}|\mathcal{W}) \right\} \tag{6.5}$$

In certain circumstances, which will be discussed below, rather than summing over all possible model sequences, only the most likely is considered. This approximation can be applied to 6.5, giving:

$$\mathcal{W}^* \simeq \underset{\mathcal{W}}{\operatorname{argmax}} \left\{ P(\mathcal{W}) \max_{\mathcal{M}} p(\mathcal{Y}|\mathcal{M})P(\mathcal{M}|\mathcal{W}) \right\} \tag{6.6}$$

Thus the quantities which must be computed for decoding are $P(\mathcal{W})$, the language model probability as defined in Section 3.3 on page 64, the acoustic model likelihood $p(\mathcal{Y}|\mathcal{M})$, and the model sequence probability $P(\mathcal{M}|W)$. Unless the lexicon allows multiple pronunciations, there will be a unique sequence of models which can be concatenated to form each word, and hence $P(\mathcal{M}|W)$ is simply unity.

The following sections will outline time-synchronous and time-asynchronous search schemes, though it is the latter which will be used to implement decoding of LDMs. Any search algorithm in its basic form should be *admissible*, meaning that it is guaranteed to find the most likely word sequence. This is not necessarily the correct word sequence, but the one which gives the highest likelihood under the acoustic and language models. An admissible search adds no errors which would not occur under an exhaustive search. However, pruning, which is the process of early removal of unlikely hypothesis in order to reduce computation, can in practice introduce some search errors.

6.1.1 Viterbi decoding

Forward dynamic programming, or Viterbi decoding as it has become known when applied to automatic speech recognition (Young et al. 2002), is a breadth-first scheme for finding the most likely path through a probabilistically weighted lattice where the axes are time and model. Breadth-first refers to a search in which all candidate hypotheses at any given time are extended before search proceeds to the next time, a characteristic which proves useful for 'on-line' speech recognition. With $\mathcal{S}_t$ representing a set of word hypotheses at time t, forward dynamic programming in its simplest form recurses in time, extending all $s \in \mathcal{S}_t$ by all models (states in the case of HMMs) which the language model allows. Therefore, the $n(\mathcal{S}_t)$ candidate hypotheses at time t become $n(\mathcal{S}_t)^{|\mathcal{M}|}$ at time $t+1$, where $|\mathcal{M}|$ denotes the number of allowable models.

The Markovian dynamics underlying HMMs provide the 'memoryless' property that all state information up to a given time is encompassed by the state at that time. An extremely useful product of this property is the Viterbi criterion (Viterbi 1967), which can be stated as:

> where two paths occupy the same state at a given time, that with the locally lower likelihood will never supersede the other.

Therefore, if the goal is to find the single most likely path through the lattice (as is frequently the case in ASR), when multiple paths share a model state at a given time, only that with the highest probability need be kept. This allows the single most likely path to be computed at far lower cost than summing over all possible paths and corresponds to making the approximation in Equation 6.6.

The search as described above is exhaustive, and often still computationally expensive despite application of the Viterbi criterion. Therefore, pruning strategies are called for, a common approach being the beam search. Under this scheme, only the best hypotheses

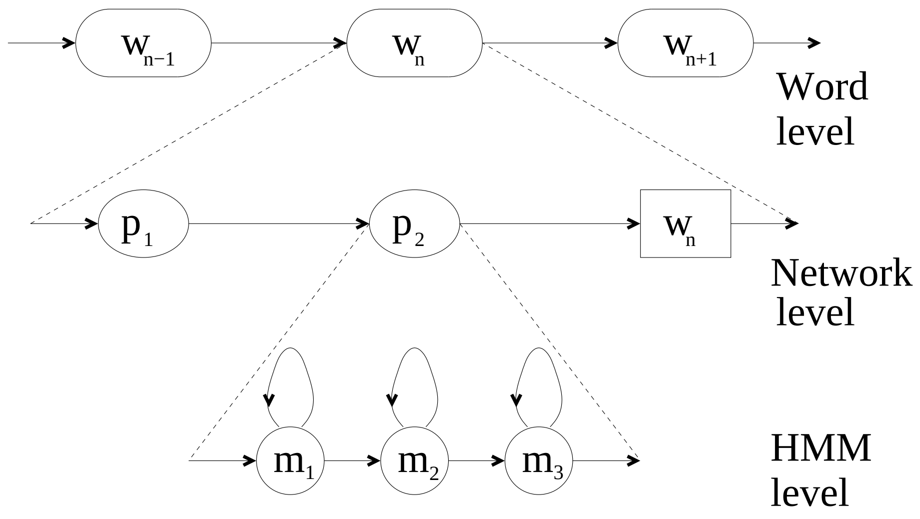

Figure 6.1: An efficient strategy for implementing Viterbi decoding for HMMs involves first decomposing all word sequences allowable under the language model into sub-word lexical items $p_1, \ldots, p_l$. These are further decomposed into a transition network consisting of set of HMM states. This figure is taken from Young et al. (2002).

are extended. A beam width Δ is chosen and any paths which have likelihoods less than $\max_{s \in \mathcal{S}_t} \log p(s) - \Delta$ are removed from the search at each time step.

An efficient strategy for implementing Viterbi decoding uses the concept of *token passing* (Young, Russell & Thornton 1989). Figure 6.1 shows how an HMM state transition network is compiled before recognition in which all word sequences allowable under the language model are decomposed into sub-word lexical items $p_1, \ldots, p_l$. These are further decomposed into a network where the nodes are HMM states. Any given node contains up to N tokens (for single-best decoding, $N = 1$), each of which stores a partial path and associated likelihood. The search moves forward in time by passing each token to each node which the transition network allows. State transition and acoustic likelihoods are added to the path likelihood contained in each token, and the N tokens with highest likelihood at each node are retained. For word decoding, transitions out of each node are recorded, and then a traceback of these records for the token with the highest path probability is used to give the most likely word sequence (Young et al. 2002).

6.1.2 A^* search

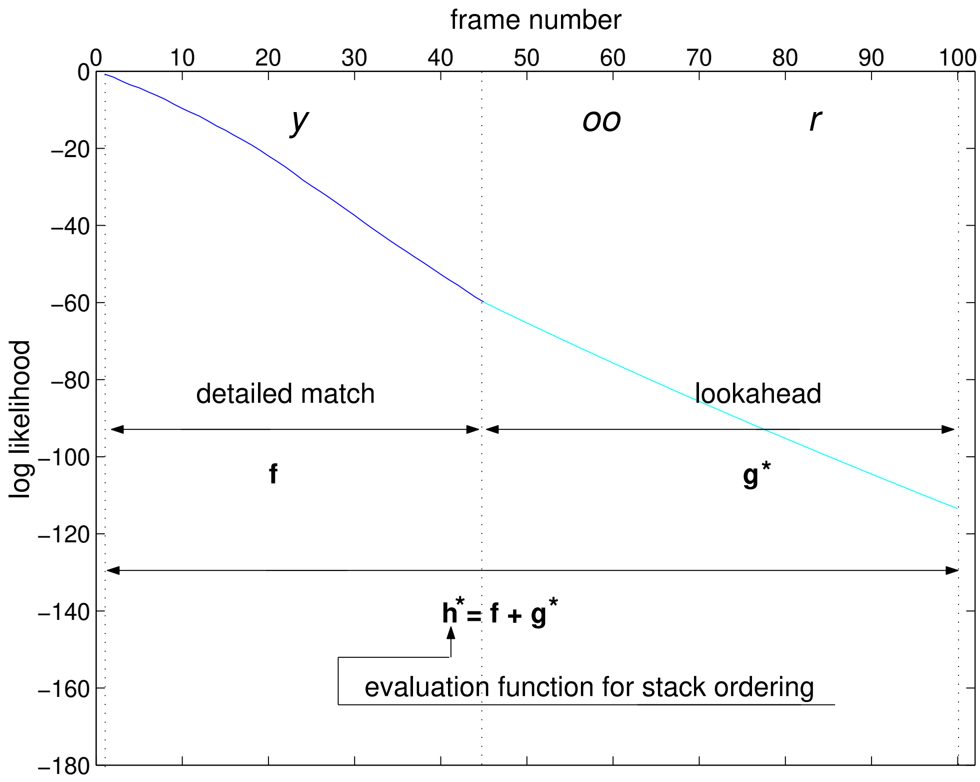

Figure 6.2: The evaluation function h^* is a combination of likelihood computed under the acoustic, language and lexical models, and an estimate of the cost of explaining the remaining observations. The lookahead function must be 'optimistic' and give an upper bound on the remaining acoustic likelihood for A^* to be an admissible search.

In a best-first search, such as A^* stack decoding, hypotheses are not explored in a time-synchronous fashion. The search order is instead determined by an evaluation function, h^*, and at each cycle it is the currently most promising partial hypothesis which is chosen for extension. For an N-frame sequence of observations $\mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, the evaluation function for a hypothesis with a path ending at time $t \leq N$ is composed of two parts:

$$h_t^* = f_t + g_t^* \tag{6.7}$$

These are the detailed match f_t which constitutes the likelihood of observations $\mathbf{y}_1^t$ under the acoustic, language and lexical models:

$$f_t = p(\mathbf{y}_1^t | m_1^{k_t})\, P(m_1^{k_t} | w_1^{j_t})\, P(w_1^{j_t}) \tag{6.8}$$

where $w_1^{j_t} = \{w_1, \ldots, w_{j_t}\}$ and $m_1^{k_t} = \{m_1, \ldots, m_{k_t}\}$ represent the hypothesised sequence of words and sub-word models respectively. The other component is g_t^* which gives an estimate of the acoustic likelihood of the remainder of the observations, $p(\mathbf{y}_{t+1}^N | \mathcal{M})$. Computing the evaluation function $h*$ from a combination of detailed match and lookahead is shown pictorially in Figure 6.2.

Using an evaluation function composed of detailed match and lookahead function is key to time-asynchronous search, as it allows the comparison of hypotheses of differing lengths. Nilsson (1971) shows that such a search as admissible as long as g_t^* gives an *upper bound* on the acoustic likelihood. Intuitively, if the lookahead underestimated the likelihood remaining, longer hypotheses would be favoured and the decoder would simply keep extending hypotheses until one was completed, without returning to explore shorter paths. Conversely, in the limiting case where $\forall t, g_t^* = 0$, decoding becomes approximately time-synchronous.

The name *stack* decoder is in fact somewhat misleading. A stack normally corresponds a last-in-first-out data structure, though in this case it comprises a list of hypotheses sorted by the evaluation function $h*$. Letting $\mathcal{S}$ represent the collection of stack items, each $s_i \in \mathcal{S}$ contains:

- a partial word sequence
- current language model state
- for each of a number of candidate end times $t_{start} \leq t \leq t_{end}$:
  - detailed match f_t
  - estimate of remaining likelihood g_t^*
  - and hence the evaluation function $h_t^* = f_t + g_t^*$.
- h^* for the hypothesis is $\max_{t_{start} \leq t \leq t_{end}} h_t^*$.

Decoding consists of alternately popping the best hypothesis off the stack, generating multiple new hypotheses by adding all allowable words, and then pushing the new hypotheses back onto the stack. One complete cycle then proceeds as follows:

- best partial hypothesis s_1 according to h^* is popped from the stack
- check if s_1 explains the entire observation sequence: if so this is the winning hypothesis and the search is finished.
- otherwise, s_1 is extended by every allowable word $w \in \mathcal{W}$ in turn to generate a new list of hypotheses $s'_1(1), \ldots, s'_1(|\mathcal{W}|)$.
- for each of $s'_1(1), \ldots, s'_1(|\mathcal{W}|)$, the likelihoods f_t, g^*_t and hence h^*_t are calculated for a range of candidate end times.
- hypotheses $s'_1(1), \ldots, s'_1(|\mathcal{W}|)$ are pushed onto the stack
- the stack is sorted, and some hypotheses may be pruned away

The first complete hypothesis to be popped from the stack is considered the winner. This can be simply extended to produce N-best lists by taking the first N complete hypotheses which are popped.

The efficiency or otherwise of an A^* search is largely determined by the lookahead function g^*. Whilst the estimate of the remaining likelihood must be optimistic, overestimates can lead to a vastly increased search space. Soong & Huang (1990) take a multipass approach where the likelihoods computed during standard forward Viterbi decoding are used to provide exact estimates of g^* for a backward A^* search which follows in order to find an N-best list of hypotheses. However, exact computation of the likelihood remaining is usually considered impractical and approximations are made using heuristic approaches. Kenny, Hollan, Gupta, Lennig, Mermelstein & O'Shaughnessy (1993) also use a multipass approach, though in this case the initial Viterbi pass comprises a less detailed search with a simplified language model to rapidly produce estimates of g^*. An alternative approach to estimating g^* is described in Paul (1992.), where the difference between the path likelihood of a given hypothesis at time t and the least upper bound of the likelihood of any hypothesis at that time is used.

Combined depth and breadth-first decoding schemes have also been investigated, such as the *start-synchronous* search of Renals & Hochberg (1995). Here, a stack ordered by an evaluation function h^* is maintained for each hypothesis end-time. Starting with the stack corresponding to the earliest reference time, a depth-first search is made in which hypotheses are either extended or pruned until the stack is emptied. Search then proceeds to the stack with the next lowest reference time. Hypotheses for which a word is successfully added are inserted in the stack corresponding to the new hypothesis end time. Such a scheme in conjunction with a hybrid connectionist/HMM acoustic model (such as described in Section 1.3.2) was shown to give near real-time decoding on a large-vocabulary speech recognition task.

6.2 Decoding for linear dynamic models

Work for this thesis has involved implementing acoustic matching for linear dynamic models within the structure of a general purpose time-asynchronous stack decoder, originally written by Simon King of the Centre for Speech Technology Research (CSTR).[1] Though the following sections deal with the case of decoding of LDMs, many of the issues involved apply to search with any segment-based model of speech. A time-asynchronous A^* search strategy was chosen for a number of reasons:

- The Viterbi criterion is not integral to the search. Such an approximation (as given in Equation 6.6) can be applied where models share the same state at some given time t. However, since LDMs have a continuous state, the Viterbi criterion can only be applied when the state is reset, which depending on the implementation may be at the ends of phones, words, or not at all. The Viterbi criterion is never admissible on a frame-by-frame basis.

- Section 4.2.3 on page 97 described how $p(\mathbf{y}_t^{t+\tau}|\Theta_m)$, the likelihood of a given model generating a sequence of observations $\mathbf{y}_t^{t+\tau} = \{\mathbf{y}_t, \ldots, \mathbf{y}_{t+\tau}\}$, is calculated. For notational simplicity, the likelihoods below are assumed to be conditioned on Θ_m.

[1] Appendix D gives a breakdown of the authorship of the various pieces of code which were required for implementation of the work in this thesis.

Once $p(\mathbf{y}_t^{t+\tau})$ has been calculated, extending acoustic matching by a single frame is straightforward. Since

$$p(\mathbf{y}_t^{t+\tau+1}) = p(\mathbf{y}_{t+\tau+1}|\mathbf{y}_{t+\tau})\, p(\mathbf{y}_t^{t+\tau}) \tag{6.9}$$

all that is required is a further forward Kalman recursion to compute $p(\mathbf{y}_{t+\tau+1}|\mathbf{y}_{t+\tau})$. However, $p(\mathbf{y}_{t-1}^{t+\tau})$ cannot be calculated in such an efficient manner. The state's initial value affects the subsequent forward filtered state statistics, and hence any likelihood computation. Therefore, a separate Kalman filter must be run to compute the model likelihoods for each candidate start time. In the light of this computational burden, the chosen search strategy must minimise exploration of unlikely paths.

- The language model contributes part of the likelihood which is used to score each hypothesis on the stack. However, unlike a Viterbi search, the language model is not used to generate each new hypothesis. Decoupling the language model and hypothesis generation in this way means that the decoder can be designed in a modular fashion. The only restriction on the language model is that it must be able to assign probabilities to initial portions of sentences consisting of whole words, for example, 'the cat sat on the'. With no requirement that the Viterbi criterion be applied on a frame level, the decoder is also flexible to the choice of acoustic model.

Section 6.1.1 above described how Viterbi decoding can be implemented for HMMs. Pre-compiling a transition network according to the language, lexical and acoustic models is a natural approach for decoding with HMMs since the models are discrete and finite-state right down to state level. However, the LDM does not fit so neatly into such a structure. LDMs give models of variable-length segments rather than frames, and with the continuous state meaning that the Viterbi criterion is inadmissible on a frame-wise basis, a time-asynchronous strategy provides a more natural and straightforward approach to implementing decoding for LDMs.

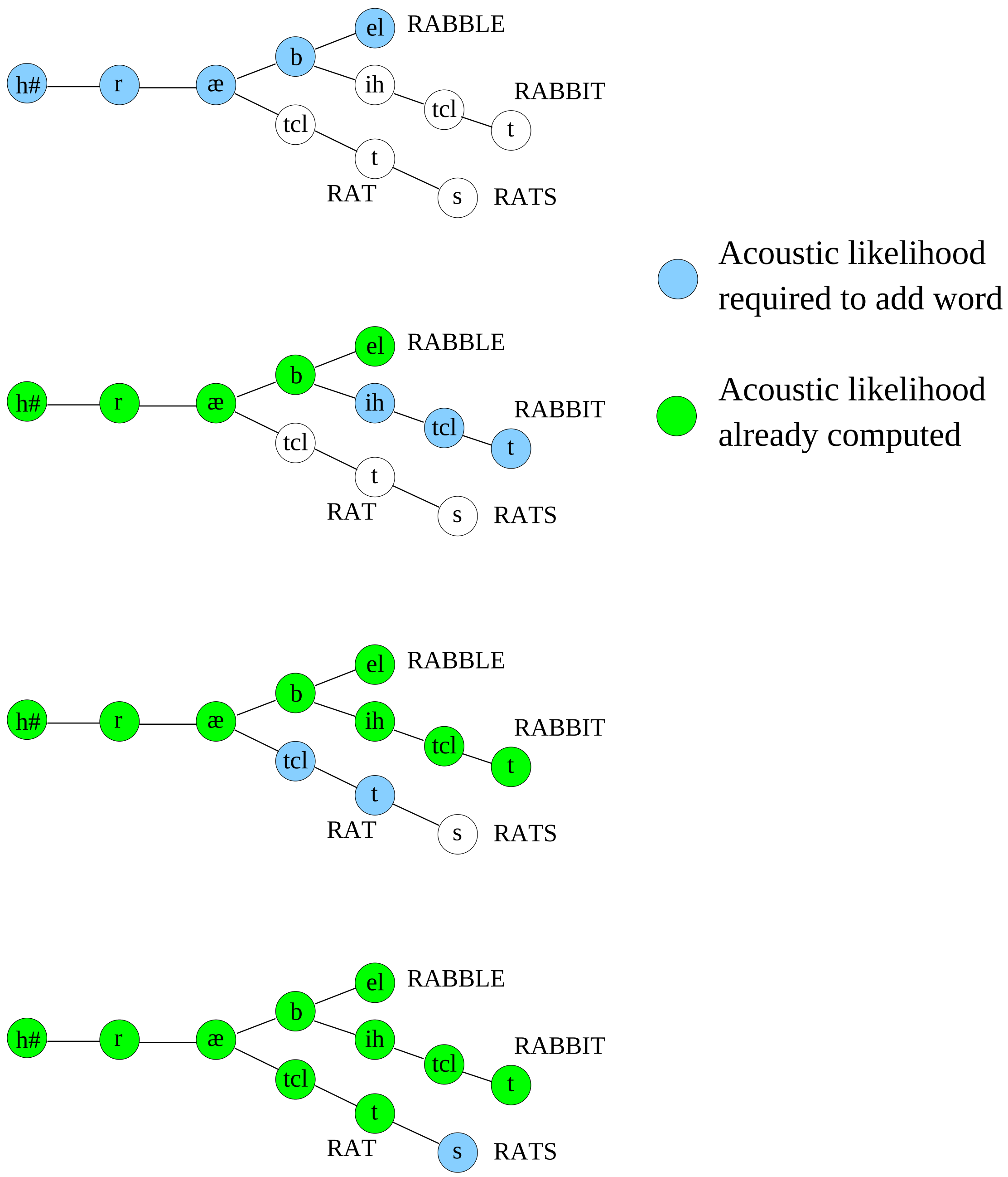

Figure 6.3: Portion of a tree-shaped lexicon. During the depth-first walk, acoustic matching for words sharing common prefixes can be shared. This figure shows the extra computation required and that which can be re-used whilst adding the words 'RABBLE', 'RABBIT', 'RAT' and 'RATS'. An optional silence model [h#] precedes each word.

6.2.1 Implementation of the core acoustic matching

For each hypothesis which is popped, decoding involves a depth-first walk over a tree-shaped lexicon as described in Renals & Hochberg (1995). Building the search around such a lexicon allows computation to be shared by paths which have common prefixes. Figure 6.3 shows a fragment of such a lexicon during the depth-first walk. Adding each new word requires some extra acoustic matching, though the figure shows how the likelihood of [`r ae b`] need only be computed once for the words 'rabble' and 'rabbit'. The experiments presented in this chapter concern phone recognition, so although a lexicon of this type will prove useful for word recognition[2], it will not form part of the decoder used in this thesis.

Acoustic matching takes place in a grid structure with time increasing down the y-axis and a column for each phone model to be added. Hypotheses are extended by whole words, one phone at a time. The detailed match for adding a new phone involves taking the likelihoods in the column corresponding to the most recently added phone, running a separate Kalman filter for each phone start-time, and entering the newly computed likelihoods in the appropriate rows of the following column. If the state is being reset between phone models, as it is for the recognition experiments reported in this thesis, the Viterbi criterion can be applied when two paths meet. This occurs where there are multiple path likelihoods to be inserted in a single grid space, and only the highest need be kept. Given that separate likelihoods must be calculated for each candidate start time, this step prevents a massive explosion in the number of hypotheses under consideration. Section 7.1.4 of chapter 7 will discuss the issues involved in continuous state decoding.

An optional silence is added at the start of each new word. The likelihoods corresponding to each of the candidate start times from the recently-popped hypothesis are entered into the first two columns of the grid. The likelihoods in the first column are picked up, and detailed match performed to compute the path likelihoods on adding a number of frames of the silence model `h#`. As usual, the Viterbi criterion is applied where paths meet, so that word-initial silence is included if any of the resulting likelihoods are higher than those of the incoming paths which already occupy the second column. This

[2]Word recognition using LDMs is a currently underway by others at CSTR.

process is shown pictorially in Figure 6.4.

Acoustic matching then proceeds with a new phone model added in each column until a word end is reached. Figure 6.5 shows the two possible final paths through the grid on adding the word ‘rat’. If this hypothesis is popped at some later cycle, there will be two candidate start times for the following word.

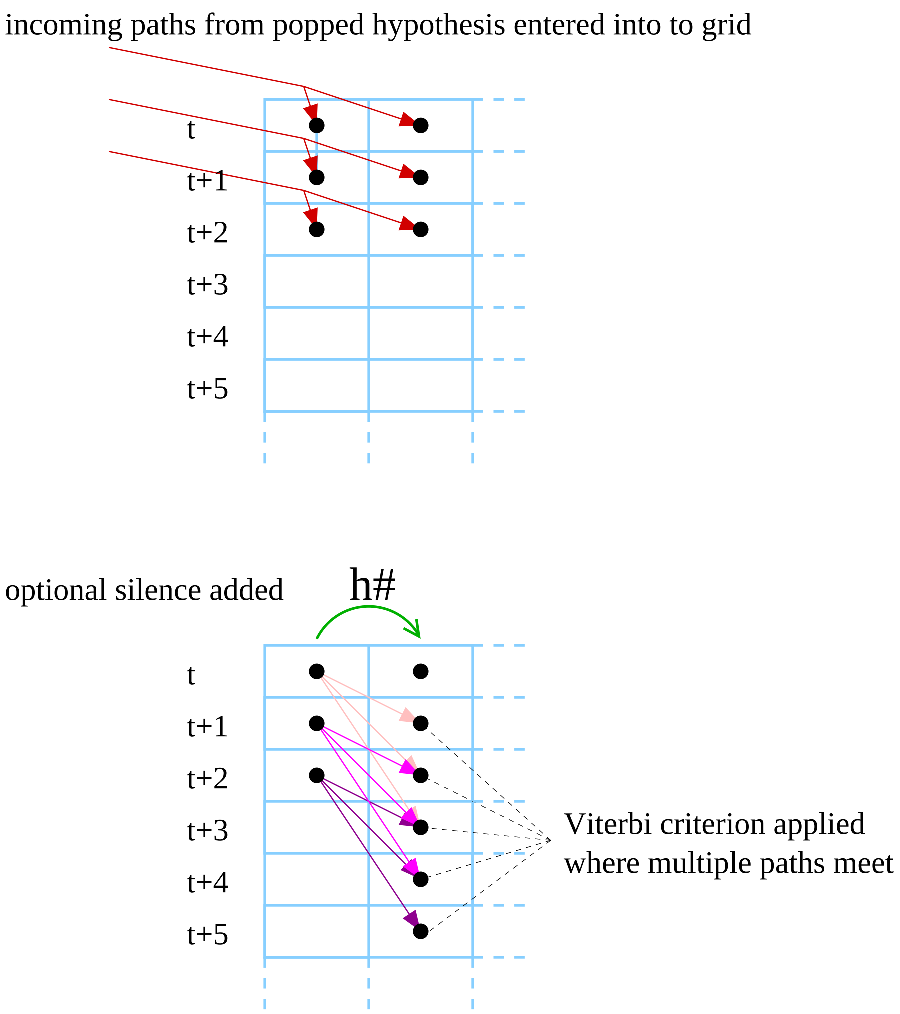

Figure 6.4: Individual words are added in a grid structure which is organised with time increasing down the y-axis and a column for each phone model to be added. To allow an optional initial silence, the likelihoods corresponding to each candidate start time are entered into the first two columns of the grid. This is shown in the upper diagram. The lower diagram shows how the likelihoods in the first column are picked up, and detailed match performed to compute the path likelihoods on adding a few frames of the silence model `h#`. A separate Kalman filter must be run for each candidate start time. The Viterbi criterion is applied where paths meet, so that the silence model is used if any of the resulting likelihoods are higher than those of the incoming paths which already occupy the second column.

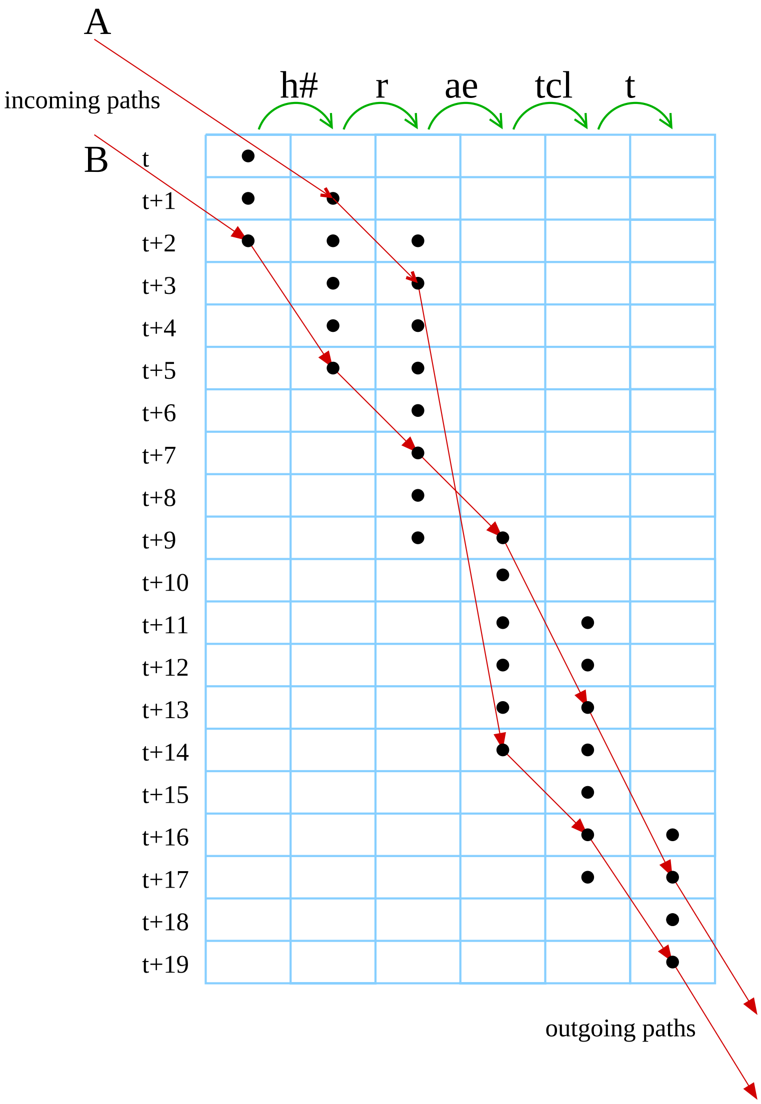

Figure 6.5: This figure shows two of the many possible paths through the grid on adding the word ‘rat’. Path A achieves this by taking a hypothesis which ends at time $t+1$ and then adding 2 frames of the `r` model, 11 frames of `ae`, 2 frames of `tcl` and then 3 frames of `t` giving a hypothesis ending at time $t+19$. Path B takes a hypothesis ending at time $t+2$ and adds 3 frames of word-initial silence before accounting for ‘rat’ using 2 frames of `r`, 2 frames of `ae`, 4 frames of `tcl` and 4 frames of `t`. Path B gives a hypothesis ending at time $t+17$ which may be picked up at subsequent cycle of the decoder.

6.2.2 Computing the lookahead function, g^*

The decoding experiments which are presented below consist of phone recognition on isolated sentences using the 61 and 46 phone model sets for TIMIT and MOCHA data respectively. For every utterance to be decoded, a Kalman filter is run across the full observation sequence for each model $m \in \mathcal{M}$. The frame-wise likelihoods under each model are ranked, then an average taken across the top n. These averages are then summed so as to produce a reverse accumulation of framewise likelihood. All experiments reported below use $n = 1$ which provides a practical upper bound on the remaining likelihood, though ignoring language model and durational constraints means that the lookahead is over-estimated.

6.2.3 Pruning

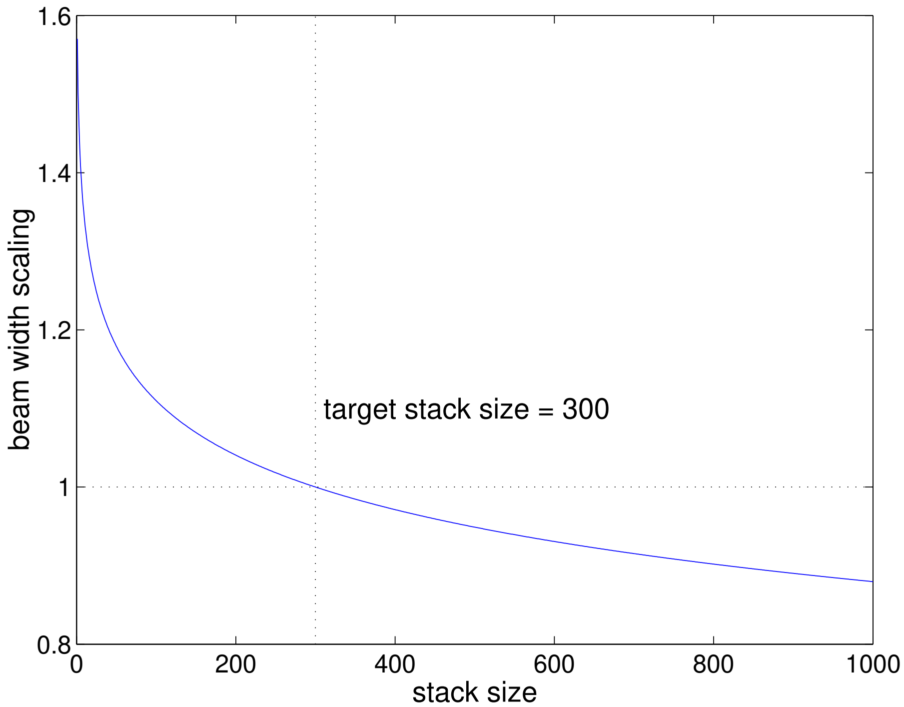

Figure 6.6: The beam width $\Delta^{(stack)}$ is updated at every iteration in order to keep to a target stack size. This figure shows the factor by which $\Delta^{(stack)}$ is scaled at every cycle where the decoder aims to maintain 300 partial hypotheses on the stack where $\alpha = 0.1$.

Pruning is implemented both in the grid and on the stack, and is dependent on calculated likelihood f_t rather than lookahead. As each word is added in turn to the most recently popped hypothesis, an upper bound $\Psi_t^{(grid)}$ is kept on the likelihoods at each time t in the grid, so that any paths for which $f_t < \Psi_t^{(grid)} - \Delta^{(grid)}$ are discarded. Similarly on the stack, an upper bound is kept for each time t, $\Psi_t^{(stack)}$. Pruning removes any hypotheses which contain no paths for which $f_t > \Psi_t^{(stack)} - \Delta^{(stack)}$.

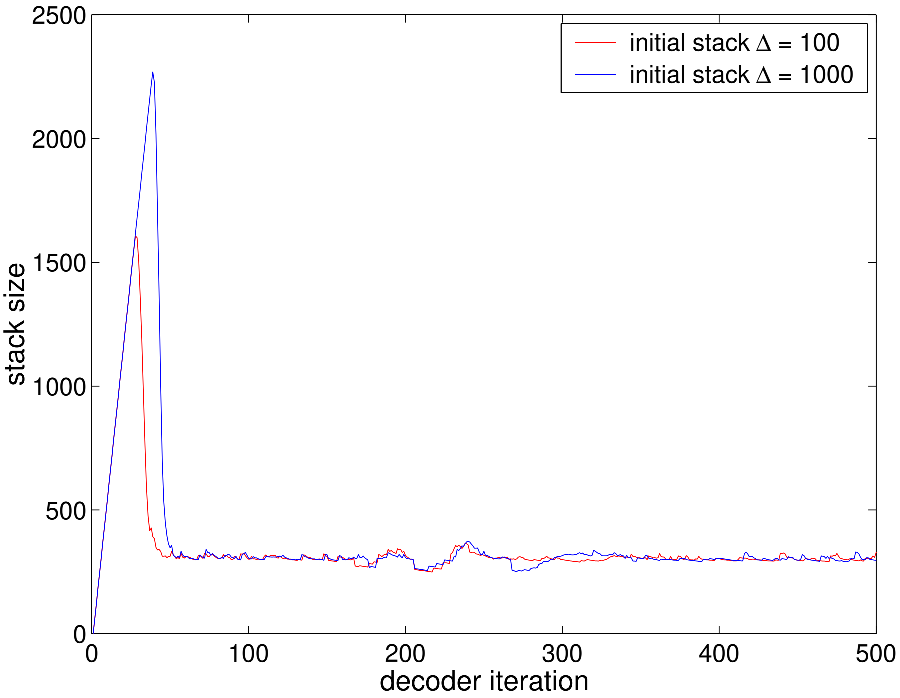

Figure 6.7: The adaptive pruning adjusts the stack beam width $\Delta^{(stack)}$ at each iteration to maintain a roughly constant number of stack items. This figure shows the first 500 cycles of the decoder for large and small original $\Delta^{(stack)}$ and a target stack size of 300.

In practice, finding suitable values of $\Delta^{(stack)}$ was a problem: tight thresholds could result in pruning away all hypotheses, whilst larger values of $\Delta^{(stack)}$ resulted in a stack which grew to a size which significantly increased decoding time. An adaptive pruning scheme was developed in which a target stack size was chosen and at each iteration, the stack beam width was updated dependent on the current stack size. Relation 6.10 gives the factor by which the stack beam width $\Delta^{(stack)}$ is adjusted:

$$\Delta^{(stack)\prime} = \left(1 - \alpha \log \frac{\text{stack size}}{\text{target stack size}}\right) \Delta^{(stack)} \tag{6.10}$$

The tuning parameter α dictates how rapidly the beam width can change. Figure 6.6

shows the factor by which $\Delta^{(stack)}$ is scaled at every cycle where the decoder aims to maintain 300 partial hypotheses on the stack. Here, $\alpha = 0.1$, the value used for experimentation. An illustration of the adaptive pruning maintaining a stack of 300 partial hypotheses whilst decoding is shown in Figure 6.7. During 1000 decoder cycles, the stack size increases initially, but is soon capped and then remains fairly constant.

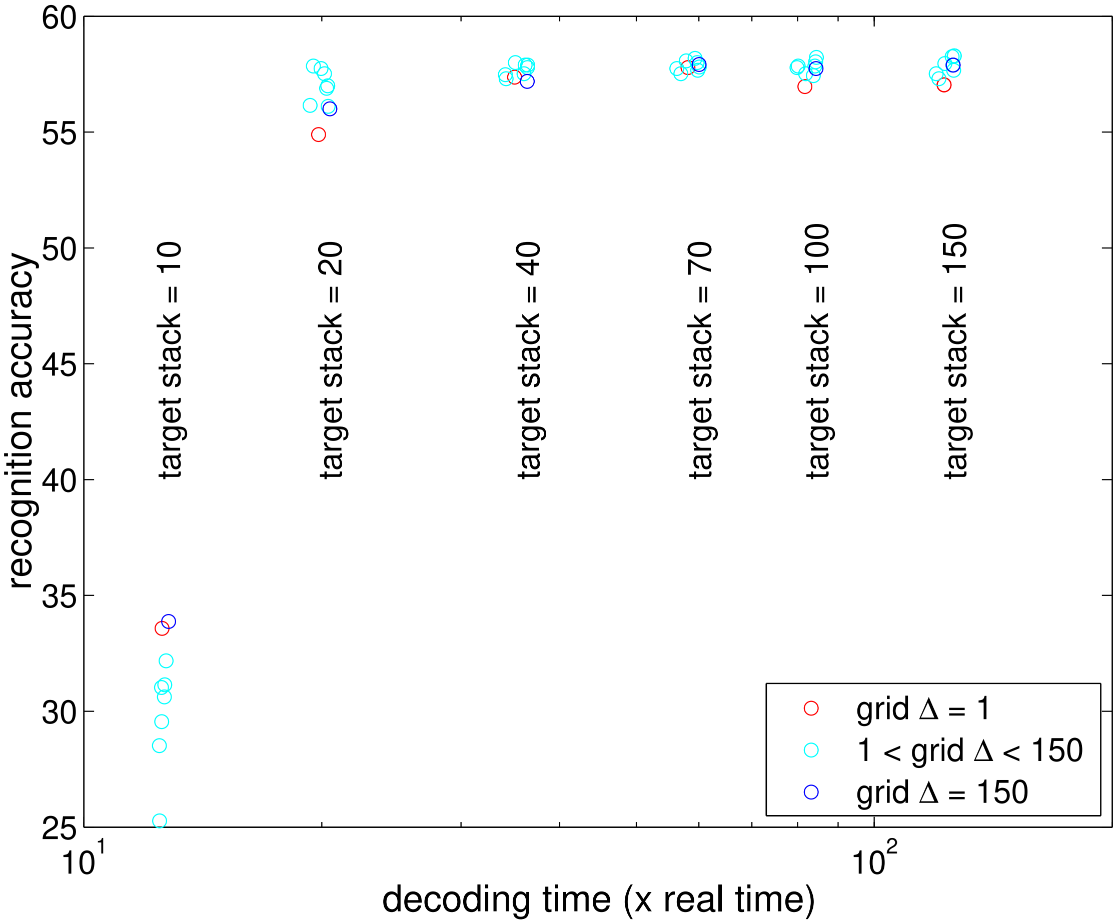

Figure 6.8: The combination of target stack size and local beam width $\Delta^{(grid)}$ affects the time taken to decode each utterance. This scatter plot shows recognition accuracy against decoding speed for target stack size ranging from 10 to 150 and $\Delta^{(grid)}$ set between 1 and 150. Each cluster corresponds to a different target stack size. With suitable parameters, decoding runs at around $20\times$ slower than real time with close to best accuracy.

To examine how pruning affects both accuracy and speed of decoding, the decoder was run over the 92 validation sentences used in the MOCHA recognition experiments with target stack sizes ranging from 10 to 150 and a grid beam width $\Delta^{(grid)}$ of between 1 and 150. The features were the 45-dimensional articulatory features used in experiments in Section 6.3.1 below. The results shown in Figure 6.8 are recognition accuracies against mean time to decode each sentence as a factor of times slower than real time, plotted on a logarithmic scale. Note that results are slightly lower than those given in Section 6.3.1 as no duration model was included.

Each cluster corresponds to a different target stack size, and the variation within each to the range of local beam widths $\Delta^{(grid)}$. It is apparent the number of partial hypotheses kept on the stack has a significant effect on the speed at which the decoder runs. The local beam width $\Delta^{(grid)}$ however affects accuracy but has little effect on time to decode each utterance. Close to highest accuracies are given with the decoder running at around $20\times$ real time. The results corresponding to the smallest and largest values of $\Delta^{(grid)}$ are shown in red and blue respectively. It is apparent that for smaller stack sizes, pruning in the grid is advantageous to recognition performance, as the highest accuracies do not correspond to the largest grid beam widths. Such pruning has the effect of removing unlikely hypotheses at the first possible opportunity.

Here, pruning in the grid is shown to make little difference to decoding speed. However it should be noted that these results are for phone recognition: it may be that the local beam width $\Delta^{(grid)}$ has a more significant effect on the decoder speed for word recognition during which multiple phone models are evaluated in the grid.

6.2.4 Efficient implementation

Pre-computation of state statistics was discussed in Section 4.2.4 on page 99. This can also be used during recognition with correspondingly significant savings. Since the state is reset between phones, computation can be further reduced by caching acoustic likelihoods. If model m with parameter set Θ_m is evaluated for the observation sequence $\mathbf{y}_t^{t+\tau} = \{\mathbf{y}_t, \ldots, \mathbf{y}_{t+\tau}\}$, the likelihoods $p(\mathbf{y}_t^t|\Theta_m), \ldots, p(\mathbf{y}_t^{t+\tau}|\Theta_m)$ are stored. Later pops of the stack will frequently require likelihoods for models starting at the same time t though with a different previous context. The cached likelihoods can be inserted into the grid without being re-computed.

6.3 Experiments

The recognition experiments use the LDM-of-phone formulation which provided the core results of chapter 5. In each case, the model set which gave the highest classification accuracy on the relevant validation set, and hence provided the final classification result, is used for recognition. The word decoder is adapted for phone recognition by treating phones as words. The lexicon simply maps each phone onto a 'word' consisting of the phone in question.

Decoding involves the combination of likelihoods from the acoustic, language and duration models. To balance the contribution of each of these, and tune decoder accuracy, a number of parameters must be set. These constitute:

- language model scaling factor. Classification experiments used simple bigram language models, though for recognition, backed-off bigrams as described in Section 3.3 on page 64 are employed.

- duration model scaling factor. The log-Gaussian duration models of Section 5.3.4 are used to aid phone recognition. Furthermore, the segment lengths at the tails of the duration distribution are disallowed by setting a minimum allowable duration model likelihood. In this way minimum and maximum segment durations are imposed.

- phone insertion penalty. Balancing the number of insertions and deletions is frequently required to maximise recognition accuracy. A phone insertion penalty is added to the acoustic log-likelihood in an effort to promote/discourage transitions from one model to the next.

- likelihood to end scaling factor. Section 6.2.2 above noted that the approach used to compute the lookahead function g_t^* would result in over-estimates of the remaining likelihood. Scaling the lookahead (by a factor < 1) might result in a faster search, though this was not used in practice. Comparison of the original estimate of the complete likelihood g_0^* and the actual computed likelihood for TIMIT recognition using δ and $\delta\delta$ parameters showed that on average, the estimate was a factor of 1.062 higher than the detailed match. This was considered to be a sufficiently low upper bound.

Unless otherwise stated, language and durational model parameters are estimated only on the relevant training sets. Scaling factors and phone insertion penalty are optimised on validation data.

Recognised output is subject to insertion and deletion errors, as well as misclassification of individual phones. The `HResults` tool distributed as part of HTK (Young et al. 2002) uses dynamic programming to align decoder transcriptions with their manual labels, before counting the number of correctly identified phones along with any insertion and deletion errors. Two statistics are frequently used to report recognition results. These are %correct and %accuracy, and are calculated as follows:

$$\%\text{correct} = 100 \times \frac{n(\text{phones correct})}{n(\text{total phone labels})} \tag{6.11}$$

$$\%\text{accuracy} = 100 \times \frac{n(\text{phones correct}) - n(\text{insertions})}{n(\text{total phone labels})} \tag{6.12}$$

It is %accuracy which gives the more reliable measure of recognition performance, as it would be possible to have a high %correct in the presence of so many insertions that the decoder output was meaningless.

Initialising state statistics

Early recognition experiments revealed that the decoder was prone to deletion errors. With the various likelihood scaling factors and phone insertion penalty optimised to give the highest recognition accuracy on a validation set, there remained many more deletions than insertions. In Section 4.2.3 on page 97, it was observed that the state covariance had an adverse effect on the classification of shorter phone segments. It was suggested that this was due to low likelihoods in the first few frames of segments prior to the state finding an appropriate location in state-space. Low likelihoods during segment-initial frames have a more significant impact on overall segment likelihood in shorter segments. Similarly, it was speculated that the large number of deletion errors during recognition were due to low likelihoods directly after state initialisation.

One method which might be used to counter this effect is by commencing Kalman filter recursions one frame prior to the hypothesised model start time, though not accumulating likelihood over the first recursion. For a model starting at time t, such a step amounts to setting $\mathbf{x}_{t-1|t-2} \sim N(\boldsymbol{\pi}, \Lambda)$, rather than $\mathbf{x}_{t|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$ as is standard. Table 6.1 gives the results of an experiment to examine if modifying the state initialisation in this way improves recognition accuracy. A subset of 120 utterances was taken from the TIMIT validation set and recognition performed with both modes of state initialisation using PLP and MFCC features, each with respective δ and $\delta\delta$ parameters. The models are the LDMs which gave the final classification accuracies of 72.2% and 72.3%, as shown in Tables 5.18 and 5.19 on pages 151 and 152 of the previous chapter, for PLPs and MFCCs respectively.

state initialised	correct	accuracy	ins	del	subs	total errors
TIMIT PLP + δ + $\delta\delta$						
$\mathbf{x}_{t\|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$	67.5%	63.5%	10.7%	34.3%	54.9%	1678
$\mathbf{x}_{t-1\|t-2} \sim N(\boldsymbol{\pi}, \Lambda)$	67.2%	62.9%	11.6%	30.1%	58.3%	1707
TIMIT MFCC + δ + $\delta\delta$						
$\mathbf{x}_{t\|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$	67.5%	63.5%	10.8%	33.6%	55.7%	1678
$\mathbf{x}_{t-1\|t-2} \sim N(\boldsymbol{\pi}, \Lambda)$	69.0%	63.6%	14.8%	27.0%	58.2%	1677

Table 6.1: Results of experiments comparing the mode of initialising state statistics for the Kalman recursions which are used to compute model likelihoods. The abbreviations ins, del and subs refer to insertion, deletion and substitution errors respectively. Initialising the Kalman recursions for model starting at time t with $\mathbf{x}_{t|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$ is standard. Alternatively, the recursions can be begun a frame prior to the hypothesised model start time, though the likelihood not accumulated over the first filter recursion. This corresponds to setting $\mathbf{x}_{t-1|t-2} \sim N(\boldsymbol{\pi}, \Lambda)$

In both cases, starting Kalman recursions a frame earlier than the hypothesised segment start does have the effect of reducing the proportion of errors which are deletions. For PLPs, the drop is from 34.3% to 30.1%, and for MFCCs from 33.6% to 27.0%. However, in the case of PLPs this is at the cost of a reduction in recognition accuracy, which falls from 63.5% to 62.9%. For MFCCs, the %correct is increased, though extra insertion errors mean that the accuracies are almost identical: 63.5% under the standard state initialisation and 63.6% when an extra Kalman recursion is included. Given that there is no evidence of benefit with MFCC features, and a reduction in accuracy using PLPs, this modified state initialisation was not adopted for recognition experiments.

Digalakis (1992), whose work is described in Section 2.3.6 on page 42, also found recognition with LDMs to be prone to deletion errors. By setting the initial state mean to be equal to the segment-initial observation, these errors were reduced, though this was only possible as the formulation of LDMs ignored subspace modelling by setting H, the state-observation mapping, to be the identity I.

6.3.1 Speaker-dependent MOCHA recognition

HMM results Wrench (2001) reports speaker-dependent recognition results on data from the MOCHA corpus using a standard HMM system. This work was mentioned on page 24 of the literature review, though is described in more detail below. HTK (Young et al. 2002) was used by Wrench to implement 3-state left-to-right triphone HMMs, the output distributions for which consisted of between 2 and 7 Gaussian mixture components. A decision tree was used for state-tying, so that $101,614$ logical models were built from between 5700 and 7000 distinct models. Recognition results were found on a 5-fold cross-validation which followed the same divisions as used in the classification experiments of chapter 5, and decoding included a phone bigram language model trained on the *full* data-set.

feature set	accuracy
articulatory	63%
acoustic	65%
articulatory + acoustic	71%

Table 6.2: Speaker-dependent MOCHA cross-validation recognition accuracies for triphone HMMs with acoustic, articulatory and combined acoustic-articulatory feature sets. Results are taken from Wrench (2001).

The recognition accuracies gained on acoustic, articulatory and combined acoustic-articulatory feature sets for speaker `fsew0` are shown in Table 6.2. Acoustic features consisted of MFCCs with δ and $\delta\delta$ parameters, and the articulatory data used the complete set of EMA, laryngograph and EPG measurements, post-processed with linear discriminant analysis (LDA). The combined features use the acoustic and articulatory features and dimensionality reduction with principal components analysis (PCA), a technique which was outlined in Section 2.1.2 on page 19. Recognition using acoustic features gave an accuracy of 65%, slightly higher (statistically significantly so) than the 63% found using articulatory features. Combining the feature sets for recognition gave the overall highest accuracy of 71%.

LDM results To allow comparison with results above, recognition experiments using LDMs on MOCHA data follow the experimental procedure used in Wrench (2001), where language model probabilities were estimated on the entire corpus, and a 5-fold cross-validation performed to accumulate recognition results for each of the 460 sentences. The scaling of the language model probabilities was set on the first cross-validation set. Recognition with LDMs uses both a language model and a duration model, parameters for which are accordingly estimated on the full training data, and scalings set on the first cross-validation set. Note that in all other experiments, the parameters of language and duration models are only estimated on training data.

For the acoustic and the combined acoustic and real articulatory features, the set of LDMs for which the highest classification accuracies were found are used for recognition experiments. For acoustic data, these are LDMs with a 9-dimensional state using MFCC and δ features. The highest accuracy with a combined feature set used LDMs with a 24-dimensional state with PLP, EMA and corresponding δ features. These classification results were originally given in Table 5.9 and Table 5.10 of chapter 5.

The actual data-set used in Wrench (2001) as described above was made available for articulatory feature recognition experiments. Classification experiments using this new feature set following the methodology of those in the previous chapter (described in Section 5.1.1 on page 115) were performed, and the highest accuracy found using an LDM with a 9-dimensional state. The full results of these exploratory experiments are given in Table E.8 on page 265 of Appendix E. A 5-fold cross-validation gave a classification accuracy of 72.4%. This represents similar performance to the experiments of chapter 5 where full articulatory features gave a classification accuracy of 72.1%.

Table 6.3 shows recognition results using LDMs on each of the acoustic, articulatory and combined acoustic-articulatory features. Acoustic-only recognition gives an accuracy

features	correct	accuracy	breakdown	
acoustic MFCC + δ	61.1%	54.4%	ins	14.6%
			del	36.6%
			subs	48.8%
			errors	6367
articulatory EMA + LAR + δ + $\delta\delta$ LDA	68.6%	60.1%	ins	21.3%
			del	27.7%
			sub	51.0%
			errors	5564
acoustic-articulatory PLP + EMA + δ	70.1%	64.4%	ins	16.0%
			del	33.2%
			sub	50.8%
			errors	4974

Table 6.3: LDM cross-validation recognition results for a 46 phone model on the `fsew0` data-set using acoustic, articulatory derived, and mixed acoustic and articulatory derived features.

of 54.4%, lower than that using the articulatory features, for which the accuracy is 60.1%. In moving from classification to recognition tasks, a greater deterioration in accuracy is found with acoustic than with articulatory features. These correspond to reductions from 75.0% to 54.4% and 72.4% to 60.1% with acoustic and articulatory features respectively. Since recognition involves jointly finding the most likely alignment and phone sequence, these results suggest that segmentations found using articulatory features lead to less confusion in phone identity. Figure 6.9 shows a confusion matrix corresponding to the recognition output using articulatory features. The vertical stripes down the mid-right of the figure shows that common errors are to misrecognize phones as the voiced fricative [`v`] and voiced oral stop [`g`].

The highest overall recognition accuracy of 64.4% is found using the combined acoustic-articulatory feature set. The breakdown of errors into insertions, deletions and substitutions included in Table 6.3 shows similar patterns for each of the feature sets. Substitutions represent around half of all errors, with deletions occurring around twice as

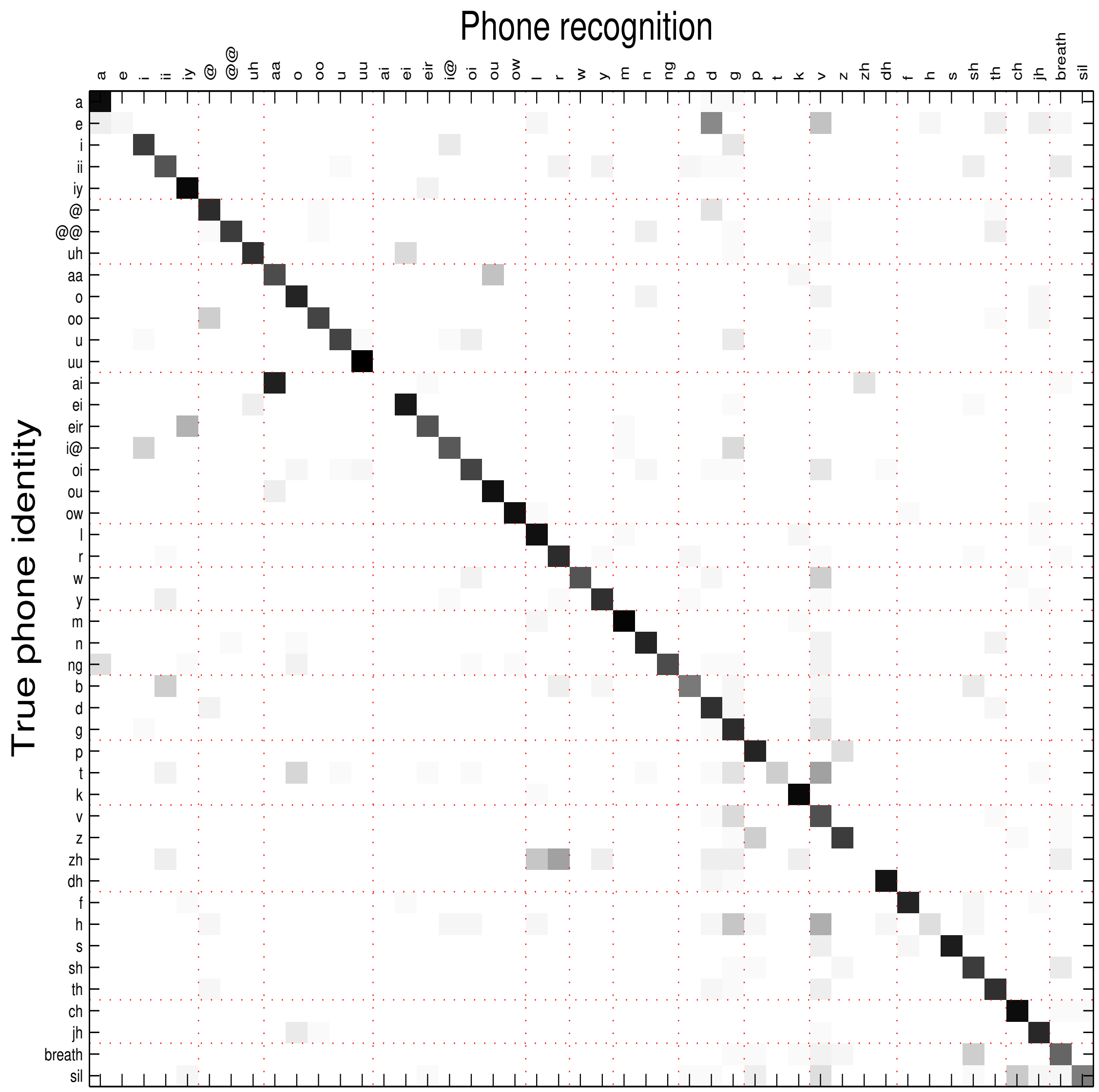

Figure 6.9: Confusion table for the recognition output with the articulatory parameters as features. The vertical stripes down the mid-right of the figure shows that common errors are to misrecognize phones as the voiced fricative [v] and voiced oral stop [g].

often as insertions. For these experiments, the decoder runs at upwards of 20× real-time, depending on the size of feature vectors and the amount of pruning. The results of 60.1% accuracy for the 45-dimensional articulatory feature set was found with the decoder running at 27× slower than real time.

These results are all lower than the HMM equivalents given in Table 6.2, though using the articulatory parameters gives the closest results. For these features, context independent LDMs give a recognition accuracy of 60.1%, which is close to the 63% found using a triphone HMM system.

Recognition experiments were also performed using the combined acoustic and network-recovered features. Rather than a 5-fold cross-validation, these use the single train/test given in Section 5.1.1 on page 116 which was used to generate the network articulation.

features	correct	accuracy	breakdown
acoustic MFCC + δ + $\delta\delta$	63.2%	57.5%	ins 13.5% del 38.1% sub 48.4% errors 620
acoustic-network articulatory MFCC + NET + δ	58.8%	55.7%	ins 6.8% del 47.1% sub 46.0% errors 645

Table 6.4: LDM recognition results for a 46 phone model on the `fsew0` data-set using acoustic and combined acoustic and network-recovered articulatory features.

Table 5.15 on page 146 gave a summary of the classification results for combinations of acoustic and network-recovered articulation, along with acoustic-only results on this same train/test division. The highest classification accuracies were 75.6% and 74.7% for acoustic-only and combined features, given using MFCCs with both δs and $\delta\delta$s, and network-recovered articulation added to MFCCs, both with corresponding δs. The results of recognition with these same models and features are given in Table 6.4, and as with classification, the highest accuracy is found with the acoustic features used alone. Adding network articulation results in a decrease in accuracy from 57.5% to 55.7%. A discussion of the possible reasons as to why the set of network-recovered articulatory features do not improve acoustic-only phone classification was given on page 147 and is equally relevant to recognition.

6.3.2 Speaker-independent TIMIT recognition

Recognition experiments were carried out for all of the TIMIT acoustic feature sets used in Section 5.2.1 of chapter 5. The model sets are those for which final classification accuracies are reported in Tables 5.18 and 5.19 on pages 151 and 152 for PLP and MFCC features respectively. As with classification, recognition is performed using a set of 61 models, though in reporting results the phone set is collapsed down to 39. The set of allowable confusions is given in 5.17 on page 150, and all results are given on the NIST core test set, which is described in Section 2.4 on page 45.

Figure 6.8 showed how recognition accuracy is affected by the level of pruning within the grid for any given target stack size. Since the lowest pruning thresholds did not necessarily yield the highest accuracies, recognition on the test set uses the same level of pruning which is applied during validation. This is necessarily fairly tight, given the number of scaling factors and word insertion penalties which must be chosen. The recognition accuracies presented in this section were produced with the decoder running between 10 and 30 times slower than real-time on a 2.4GHz Pentium P4 processor, depending on the dimension of the feature vector.

Recognition results are given in full in Table 6.5. For PLP cepstra, adding δs increases recognition accuracy from 55.2% to 58.5%, and has a balancing effect on the occurrence of deletion and insertion errors with the ratio shifting from over 5:1 to around 3:1. However, further adding $\delta\delta$s gives no extra increase in accuracy, and in fact gives a reduction in % correct. The recognition experiments which use MFCC features show that accuracy is improved on adding δs and further including $\delta\delta$s. Recognition accuracy increases from 51.1% to 57.2% on adding δs. The number of errors is reduced from 3588 to 3140, and the percentage of these which are deletions falls from 45.5% to 41.5%. Also then adding $\delta\delta$s gives the highest overall accuracy of 60.3%, and further balances the occurrence of insertions and deletions.

features	correct	accuracy	breakdown	
PLP	58.7%	55.2%	ins	7.7%
			del	39.2%
			sub	53.1%
			errors	3283
PLP + δ	62.9%	58.5%	ins	10.6%
			del	32.7%
			sub	56.7%
			errors	3045
PLP + δ + $\delta\delta$	62.0%	58.5%	ins	8.3%
			del	37.0%
			sub	54.7%
			errors	3042
MFCC	54.0%	51.1%	ins	5.9%
			del	45.5%
			sub	48.6%
			errors	3588
MFCC + δ	60.0%	57.2%	ins	6.6%
			del	41.5%
			sub	51.8%
			errors	3140
MFCC + δ + $\delta\delta$	63.9%	60.3%	ins	9.3%
			del	35.0%
			sub	55.8%
			errors	2914

Table 6.5: Speaker-independent recognition results using acoustic data from the TIMIT core test set. Results correspond to the 39 phone set.

Figure 6.10 shows a confusion table corresponding to the highest TIMIT recognition result of 60.3% using MFCCs with δ and $\delta\delta$ parameters. The majority of the confusions appear to be between vowels, with phones commonly misclassified as [`ix`] [`ax`] or [`ao`]. Also, errors appear in making voicing decisions, with [`b, d`] being frequently recognised as their voiceless equivalents [`p, t`].

Context-independent TIMIT phone recognition results are given by Goldenthal (1994), whose work is summarised on page 44 of the literature review. A 39-phone recognition accuracy of 61.9% was found on the core test set, and increased to 63.9% by incorporating explicit models of phone transitions. These results used gender-specific models, and assume that the gender of a given speaker is known[3].

To allow comparison with this result, gender-specific experiments were also performed with the LDM-of-phone formulation using MFCCs with δs and $\delta\delta$s. Classification followed the methodology of Section 5.2.1 on page 149, except that separate model sets were trained, validated and tested for male and female speakers. An overall classification accuracy of 73.6% was found, which represents a statistically significant increase over the original highest LDM-of-phone classification accuracy of 72.3%, given in Table 5.19 on page 152. Gender-dependent recognition on the TIMIT core test set gave an accuracy of 61.5%, shown in Table 6.6. This constitutes a statistically significant increase on the original gender-independent model accuracy of 60.3%, and is close to Goldenthal's 61.9% for which explicit transition models were not used.

model	recognition accuracy
gender-independent	60.3%
gender-dependent	**61.5%**

Table 6.6: Gender-dependent recognition on the TIMIT core test set gave an accuracy of 61.5%, a statistically significant increase on the original gender-independent model accuracy of 60.3% and close to Goldenthal's 61.9%.

[3]Lamel & Gauvain (1993*b*) showed that gender decisions can be reliably made from speech parameters, findings backed up by Goldenthal (1994) who reports that of 250 utterances from the TIMIT test set, gender classification gave an accuracy of 100% using the trajectory models as described on page 44 of this thesis.

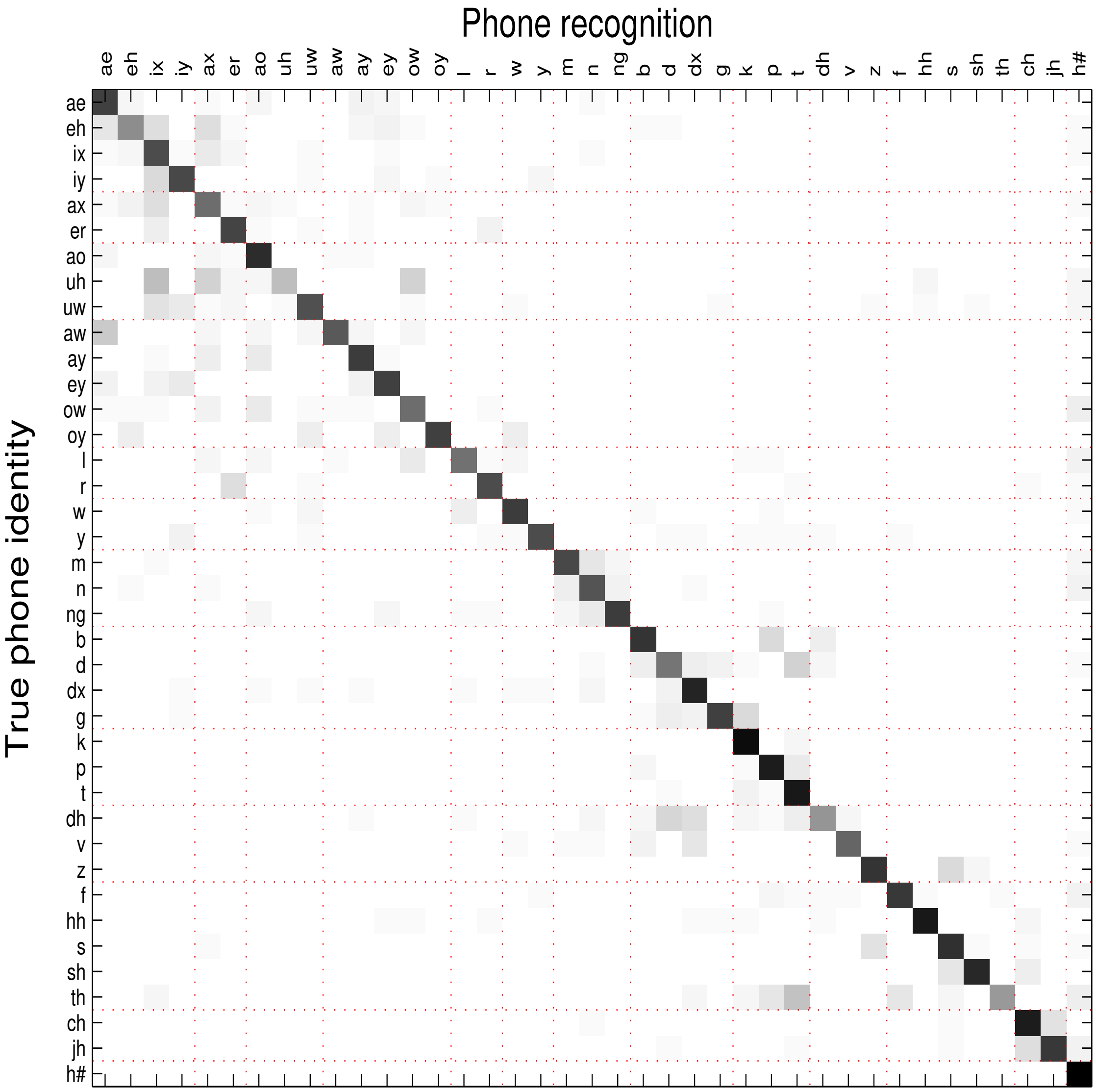

Figure 6.10: Confusion table for LDM recognition output with MFCCs + δs + $\delta\delta$s as features. The majority of the confusions appear to be between vowels, with phones commonly misclassified as [ix] [ax] or [ao]. Errors also appear in making voicing decisions, with [b, d] being frequently recognised as their voiceless equivalents [p, t].

Chapter 7

Conclusions and future work

7.1 Analysis of results

On page 11 of the introduction, it was stated that the work in this thesis is motivated by the belief that a model which reflects the characteristics of speech production will ultimately lead to improvements in automatic speech recognition performance. The following sections describe how this has been interpreted in practice, and analyses the results of the investigations reported in previous chapters. Section 7.3 on page 234 then goes on to describe how this work will be extended in the future.

7.1.1 Modelling of inter-frame dependencies

Computing the likelihood of a sequence of feature vectors within a single HMM state $q_j \in \mathcal{Q}$ is invariant under random reorderings of the data.[1] Speech is in fact a highly ordered signal (where the ordering itself carries information), due to the constrained nature of the production mechanism. Furthermore, given that features are extracted within overlapping windows on the acoustic signal, successive frames of data are destined to be highly correlated. Incorporating the ordering present in the parameterized speech signal with an explicit description of the inter-frame dependencies should lead to improved modelling.

[1] This statement assumes that, if included, any derivative information in the form of δs and $\delta\delta$s has already been appended to the feature vectors. Clearly reordering the data prior to calculation of δs and $\delta\delta$s will yield a distinct set of features.

Motivation for applying LDMs Section 3.4 on page 66 described experiments in which the ability of linear and non-linear models to describe the dependencies present within and between phone segments was compared. Within segments, the preceding frame was found to be a good predictor of the segment-central frame, with regressions on TIMIT acoustic data accounting for around 70% of the variation in the dependent variable. With the segment-initial frame as predictor, regressions gave poorer fits, explaining between 40% and 45% of the variation. Given the overlapping nature of feature extraction, and variation due to context, it was expected that using the preceding frame as predictor would lead to better absolute fits than the segment-initial one. However, in both cases, linear models gave fits which were close to those of their non-linear counterparts. Between segments, not only were the regressions less able to explain the data, but the effect of crossing phone boundaries was shown to be more detrimental to the performance of linear than for non-linear predictors.

In summary, much of the variation within segments could be explained by the preceding frame, and a linear predictor gave over 96% of the fit of a non-linear model for each of TIMIT MFCCs and PLP cepstra. However, between segments, regressions could explain less of the variation and linear models were significantly outperformed by non-linear models. These findings give motivation to the application of first order linear model of speech parameters *within* but not between phone segments.

Contribution of the dynamic state process The bulk of the classification experiments of Chapter 5 were intended to assess the contribution made by the addition of a model of inter-frame dependencies. Linear dynamic models were compared with otherwise equivalent static models, and the addition of modelling of temporal correlations was found to give modest yet statistically significant increases in classification accuracy. Table 7.1 shows the highest accuracies using static and dynamic models on the TIMIT corpus, with LDMs applied in their fully parameterized form and one model per phone class. Classification using LDMs gave an accuracy of 72.3%, a 3.5% relative error reduction over the best result found using a static model of 71.3%. A paired t-test showed this result to be statistically significant which demonstrates that the increase is consistent over the test data.

model	classification accuracy	recognition accuracy
static model	71.3%	58.1%
LDM-of-phone	**72.3%**	**60.3%**

Table 7.1: Comparison of the highest accuracies using static models and LDM of phone models. Features are TIMIT MFCCs with δs and $\delta\delta$s.

Recognition using LDMs on the TIMIT core test set gave an accuracy of 60.3%, which is higher than the recognition accuracy of 58.1% found with an otherwise equivalent static model[2]. This represents a relative error reduction of 5.3%, and again is shown to be significant using a paired t-test. Therefore, for models of complete phone segments, accounting for inter-frame dependencies is shown to give acoustic modelling which results in more accurate phone identity decisions.

Experiments later in the chapter used multiple regime (MR) LDMs in which phone segments are split into a number of regions, each one modelled by an LDM. Table 7.2 shows the highest results found using multiple regime static and dynamic models on TIMIT MFCCs used alone and with both δs and $\delta\delta$s. These results given here are for the MR LDMs in which the state is passed between sub-models. Also shown are results using the standard LDM-of-phone models and an accuracy gained by combining static and dynamic models.

With MFCCs used alone, the multiple regime models give statistically significant performance increase over both standard LDM-of-phone and the MR static models. When both δ and $\delta\delta$ parameters are included in the features, the MR LDM still gives a higher accuracy than the MR static model and standard LDM, though the increase is not statis-

[2] Many thanks to Simon King for preparing this baseline result. HTK (Young et al. 2002) was used to implement one-state monophone HMMs with single full covariance Gaussian output distributions. Models were trained, validated and tested on identical data to that used in all LDM recognition experiments. The language model was also the same. The validation set was used to select the language model scale factor, word insertion penalty, and a beam pruning width such that there were minimal search errors. The best models were initialised with uniform segmentation and Viterbi training (`HInit`), then Baum-Welch to convergence with fixed segment label times (`HRest`) followed by a single iteration of full embedded training(`HERest`). It should be noted that the HMM implementation of the static models means that the duration model will be exponential rather than log-Gaussian as used in LDM recognition.

classification accuracy		
model	MFCC	MFCC + δ + $\delta\delta$
LDM-of-phone	67.4 %	72.3%
MR static	68.6 %	74.3%
MR LDM	**69.5 %**	74.5%
MR combined LDM and static	–	**74.9%**

Table 7.2: Comparison of accuracies of static and dynamic multiple regime models. Standard LDM-of-phone results are also given for reference. The features are TIMIT MFCCs either used alone or with δ and $\delta\delta$ parameters. Bold face denotes an accuracy which is statistically significantly higher than the others using the same features. Results were originally given in Table 5.28 on page 167.

tically significant in this case. When the likelihoods under the two models are combined, a classification accuracy of 74.9% is found, which represents a statistically significant increase over either of the models used individually. There are two main conclusions to be drawn from these results.

Firstly, the distribution of errors under static and dynamic models are sufficiently different that accuracy can be improved using combinations of the two. However, LDM estimation should allow modelling of static distributions, as a zero or near-zero state evolution matrix F would remove or significantly reduce the dependencies between successive frames. Assuming parameters can be reliably estimated, overall static model performance should never exceed that found using LDMs, though this is complicated by discrimination amongst models. Figure 5.23 on page 169 gave a comparison of the classification accuracies under the MR LDM and MR static models broken down by phonetic category, and shows that static models gave higher accuracy for diphthongs. This class of phones are characterised by spectral transitions, and it was expected that a dynamic model would give an advantage in this case. Closer inspection of the results shows that whilst static models correctly classified a greater number of diphthong segments, there were also almost 50% more phones misclassified as diphthongs by the static models than by dynamic models.

The second conclusion which can be drawn is that partitioning phone segments into regions each of which is modelled by an LDM improves classification accuracy. However, this result comes accompanied by a caveat: the MR LDMs only give real performance improvements over MR static models where MFCC features are used alone. The inclusion of δ and $\delta\delta$ parameters reduces the benefit of adding a dynamic state process. Section 5.3.2 on page 164 made the point that sub-dividing phones in this way risks that modelling tends toward that of an HMM, where each model describes a short, stationary region of the parameterized speech signal. With sub-phone regions frequently consisting of only a few frames, each model is trained on a set of relatively short time-series from which any underlying dynamics must be extracted. The results in Table 7.2 do not justify the extra computation involved in applying MR LDMs over MR static models. Section 7.3.1 below discusses the sub-optimality of this implementation of MR LDMs and points at a future direction intended to realise the potential of such models.

7.1.2 Modelling spatial dependencies

Feature extraction for ASR, as described in Section 3.1.2 includes steps which are designed to decorrelate the final parameters. However, dependencies persist between feature dimensions. A model which can account for these spatial correlations should have an advantage over one in which they are ignored. Section 5.2.4 on page 157 gave the results of classification on MOCHA and TIMIT data for LDMs on which a variety of constraints had been placed. The results using TIMIT MFCCs with δ and $\delta\delta$ parameters are summarised in Table 7.3 for variations on fully parameterized LDMs which alter the modelling of spatial dependencies.

The function of the matrix H is to allow the observation and state distributions to occupy distinct vector spaces. Frequently, fewer degrees of freedom are needed to describe a process than present in the parameters which describe it, and H offers the chance to give a compact representation of the underlying dynamics. Setting H to be the identity I casts the LDM as a smoothed Gauss-Markov model and removes the capacity for subspace modelling. Table 7.3 shows that the classification accuracy using such a model is 71.7%, which represents a statistically significant reduction on the 72.3% given by a fully parameterized LDM. The linear dimensionality reduction which H provides is demonstrated here to yield a modest yet consistent performance improvement. The figures in the third column of Table 7.3 show that a full H matrix results in the models using many fewer parameters, 83K compared to the 243K when H is set to be the identity I. This arises as 39×39 rather than 9×9 state evolution, noise and initial covariance matrices, F, D and Λ respectively, must be estimated.

model	classification accuracy	total parameters
H identity	71.7%	243K
C, D diagonal	69.9%	36K
D diagonal	72.2%	81K
full LDM	72.3%	83K

Table 7.3: Comparison of accuracies using LDMs with a variety of constraints which affect modelling of spatial correlations for TIMIT MFCCs with δs and $\delta\delta$s. Note that the initial state covariance was not restricted during these experiments. The total number of parameters for each LDM variant is given also.

Setting both observation noise C and state noise D to be diagonal is the only one of these variations which represents a theoretical loss of generality. It was shown in Section 4.1.2 on page 87 that with $\mathbf{x}_t$ and Σ_t representing the state mean and covariance at time t, the output distribution is:

$$\mathbf{y}_t \sim N(H\mathbf{x}_t + \mathbf{v}, H\Sigma_t H^T + C) \tag{7.1}$$

With C and D diagonal, 7.1 gives a model of the spatial dependencies where the correlation structure of the data is absorbed into H, so that the observation noise covariance is approximated by a projection of the lower-dimensioned state error distribution. The classification accuracy using LDMs with diagonal covariances for both state and observation is 69.9%, which is statistically significantly lower than all of the other results in Table 7.3 where output noise distributions are described fully. These results demonstrate that detailed modelling of spatial dependencies is advantageous in making phone-class decisions.

The classification accuracy of 72.2% where just the state covariance has been set to be diagonal is almost identical to that found with a fully parameterized LDM. The figures in

Table 7.3 demonstrate that the transformation H can provide the rotations required to map the state space onto a basis in which the dimensions are independent with minimal loss of accuracy and a slight reduction in parameterization.

7.1.3 Incorporation of articulatory information

Along with comparing the performance of static and dynamic models, the classification experiments of Chapter 5 examined the effect of adding articulatory features to acoustic parameters. The difficulties inherent in measuring human articulation mean that real articulatory data can only be used as a design tool. However, with building a model which reflects the properties of speech production cited as a goal, an examination of the properties of real articulatory parameters provides an ideal starting point.

The regressions of Section 3.4, which compare linear and non-linear models of the dependencies between and within phone segments, find both the overall highest absolute R^2 values and closest linear/non-linear performances on MOCHA EMA data. With the preceding frame as the predictor, a linear regression gives an R^2 of 0.981, which is 99.4% of that of the non-linear model. These results demonstrate the strong correlation which exists between articulatory feature vectors spaced 10ms apart. This is expected given the relatively slowly varying nature of the speech production mechanism.

The regression models were also able to give good predictions of the data when the dependent and explanatory variables were spaced a number of frames apart. Much of the variation in the segment-central frame – almost 80% – could be explained using a linear model with the segment-initial frame as predictor. Crossing phone boundaries was found to have minimal impact on the performance of the non-linear regressor, though lead to a slight reduction of that of the linear model. These results serve to demonstrate the long-range predictability of articulatory features, even though the inter-segmental dependencies may be better modelled with a non-linear predictor.

Table 7.4 gives a summary of the best classification and recognition results for acoustic, articulatory and combined acoustic-articulatory features for the MOCHA corpus. Acoustic and articulatory features give different distributions of errors, as shown pictorially in Figure 5.11 on page 136. This is also apparent in moving from classification to recognition tasks: whereas classification with acoustic data gives a higher accuracy than that with articulatory parameters, 75.0% compared to 72.4%, recognition with articulatory data gives greater accuracy than that with acoustic features, 60.1% compared to 54.4%. Given that the difference between classification and recognition tasks is in finding the alignment of segments, these results suggest that articulatory features allow the set of LDMs to find phone boundaries more accurately.

features	classification accuracy	recognition accuracy
acoustic	75.0%	54.4%
articulatory	72.4%	60.1%
acoustic-articulatory	**79.2%**	**64.4%**

Table 7.4: Comparison of the highest classification and recognition accuracies on MOCHA data for acoustic, articulatory and combined acoustic-articulatory feature sets. These experiments use measured articulation from the MOCHA corpus.

Adding articulatory information to acoustic features gives significant performance improvements, with classification accuracy raised from 75.0% to 79.2% and recognition accuracy from 54.4% to 64.4%. These results demonstrate that the information present in articulatory data can be used to supplement that of the parameterized acoustic signal, giving improved ability to discriminate between phones.

The question remains as to how these findings can be used to improve a practical system for which measured articulation is not available. Experiments which replaced measured articulation with network-recovered parameters were not able to replicate these results. In certain cases, classification performance was improved, though the overall highest accuracy was found using only acoustic features. Table 7.5 gives a summary of the best acoustic-only and combined acoustic and network-recovered articulation classification results and their corresponding recognition accuracies. In both cases, addition of network-recovered articulatory parameters leads to a slight reduction in the system performance, 75.6% to 74.7% for classification and 57.5% to 55.7% during recognition.

At each time, given a context window on the acoustic parameters, the MLPs which produced the parameters used here give an averaged articulatory configuration. Such a mapping may be producing articulation which is consistent and therefore predictable for

features	classification accuracy	recognition accuracy
acoustic	75.6%	57.5%
acoustic-articulatory	74.7%	55.7%

Table 7.5: Comparison of the highest classification and recognition accuracies on MOCHA data for acoustic and combined acoustic-articulatory feature sets. These experiments use the network-recovered articulation produced to accompany MOCHA data.

articulators which in measured data show a great deal of variation. Not all articulators are critical for the production of each phone (see page 51 for a description of critical/non-critical articulators), and Section 4.4.2 on page 108 showed that the variance captured in the observation noise C for different feature dimensions reflected notions of which articulator would be critical in producing the consonantal phones. These variances are used to weight the contribution of each feature dimension in computing the likelihood of a model given a sequence of observations. It seems likely that for network-recovered articulatory features, there is an overemphasis of the contribution of non-critical articulators in computing likelihoods.

Section 5.1.6 on page 147 suggested that an alternative type of neural network may provide an articulatory inversion mapping which is more suitable for the purposes of ASR. An approach taken in Zacks & Thomas (1994) replaces the sum of squares error function with one which incorporates a measure of correlation. The intention is to push learning toward recovering the shape of articulatory trajectories, rather than simply estimating the conditional mean. Alternatively, work reported in Richmond (2001) uses mixture density networks which allow a multimodal output distribution. Such a network, with a single mixture component, would estimate the variance of each articulatory dimension along with its conditional mean. Incorporating this measure into the LDM's observation noise covariance would provide a measure of the criticality or otherwise of a given articulator, by weighting its contribution in any likelihood calculation. Real articulation is shown to provide cues which are not as apparent in acoustic data. Ultimately, incorporating artificially recovered articulatory parameters will only provide performance improvements if the inversion mapping can reliably derive these cues from the acoustic signal.

7.1.4 Continuous underlying representation

Building a continuous underlying representation of the parameterized speech signal ties in closely with modelling of temporal dependencies. A state process which is continuous across segment boundaries brings the possibility of modelling longer-range dependencies than are contained within phone segments whilst also reflecting the constrained nature of the production mechanism. Section 5.4 presented a number of classification experiments in which the state statistics were passed across segment boundaries. The results are summarised in Table 7.6.

state	classification accuracy
MOCHA EMA	
reset	59.5%
passed	57.0%
TIMIT MFCC	
reset	67.4%
passed	67.0%

Table 7.6: Comparison of classification accuracies in which the state is either reset or passed between phone segments. Results are given for MOCHA EMA and TIMIT MFCC features.

For both MOCHA EMA data and TIMIT MFCCs, the classification accuracy decreases when the state statistics $\mathbf{x}_{t|t}$ and $\Sigma_{t|t}$ are passed across phone boundaries. Rosti & Gales (2003) report preliminary results for a similar experiment using LDMs in which it was also found that a fully continuous state gives a slight reduction in performance. A factor-analysed hidden Markov model (FAHMM) (Rosti & Gales 2002) was used to generate 50-best lists for utterances from the resource management (RM) (Price et al. 1988) test set. Rescoring using LDMs gave a word error rate of 11.00% where the state was reset between phone segments, and 11.82% where the state was continuous.

The initial results both in this thesis and from Rosti & Gales (2003) are not conclusive for this implementation of LDMs, as the success of such an approach might depend on occasional resetting of the state. There is a great deal of variation in the nature of the transitions between segments. In some cases, these will be highly non-linear, such as found between the closure and release portions of plosives where the change point is defined by an abrupt shift in the spectral energy. At other times, the segmental boundaries are less well-defined, such as in the transition between a vowel and a nasal stop, where anticipatory nasalisation colours the vowel sound and the spectral transitions are smooth. It may be that resetting the state for the first of these examples would act as a regularizer for the state covariances, but allowing passing of the state in the second would enhance modelling.

Figure 7.1 shows some of the statistics computed during a forward filter recursion across a complete utterance from the TIMIT corpus, both with the state passed across model boundaries, and also where it has been reset. The framewise likelihoods are plotted, along with the log determinants of the error covariance $\Sigma_{\mathbf{e}_t}$ and state covariance $\Sigma_{t|t}$. The same set of models was used to generate both state-passed and state-reset plots so that the effect of resetting can be seen in isolation. The true models according to the manual labels were used for each segment, having been trained on the full TIMIT training data in the normal way with the state reset at the start of each segment.

When the state is reset to have its initial distribution $N(\boldsymbol{\pi}, \Lambda)$ between segments, phone boundaries are evident as there are sudden reductions in the magnitude of the state covariance $\Sigma_{t|\,t}$ (shown in the 3^{rd} and 4^{th} plots on different y-scales) at these points. For longer segments, the covariance appears to converge to a set magnitude regardless of whether the state has been reset or not.

With the state continuous across segment boundaries, estimates of the state covariance $\Sigma_{t|\,t}$ at the start of phones tend to have larger magnitudes than when steady-state values are reached. Conversely, if reset at the start of segments, the magnitude of the state covariance is dramatically under-estimated. Training of LDMs was described in Section 4.2.2 and estimation of the initial state covariance follows that found in Digalakis et al. (1993), Ostendorf, Digalakis & Kimball (1996) and Roweis & Ghahramani (1999). However, Ghahramani & Hinton (1996a) give a form which adds the covariance of the initial smoothed state vectors about the initial state mean $\boldsymbol{\pi}$ to the estimate of Λ. With $\hat{\mathbf{x}}^{(k)}_{1|N_k}$ representing the initial smoothed state vector for the k^{th} of K sequences which has N_k frames, the extra term is $\sum_{i=1}^{K}(\mathbf{x}^{(i)}_{1|N_i} - \boldsymbol{\pi})(\mathbf{x}^{(i)}_{1|N_i} - \boldsymbol{\pi})^T$. This modification would certainly result in larger estimates of the initial state covariance, and might provide a useful addition for future work.

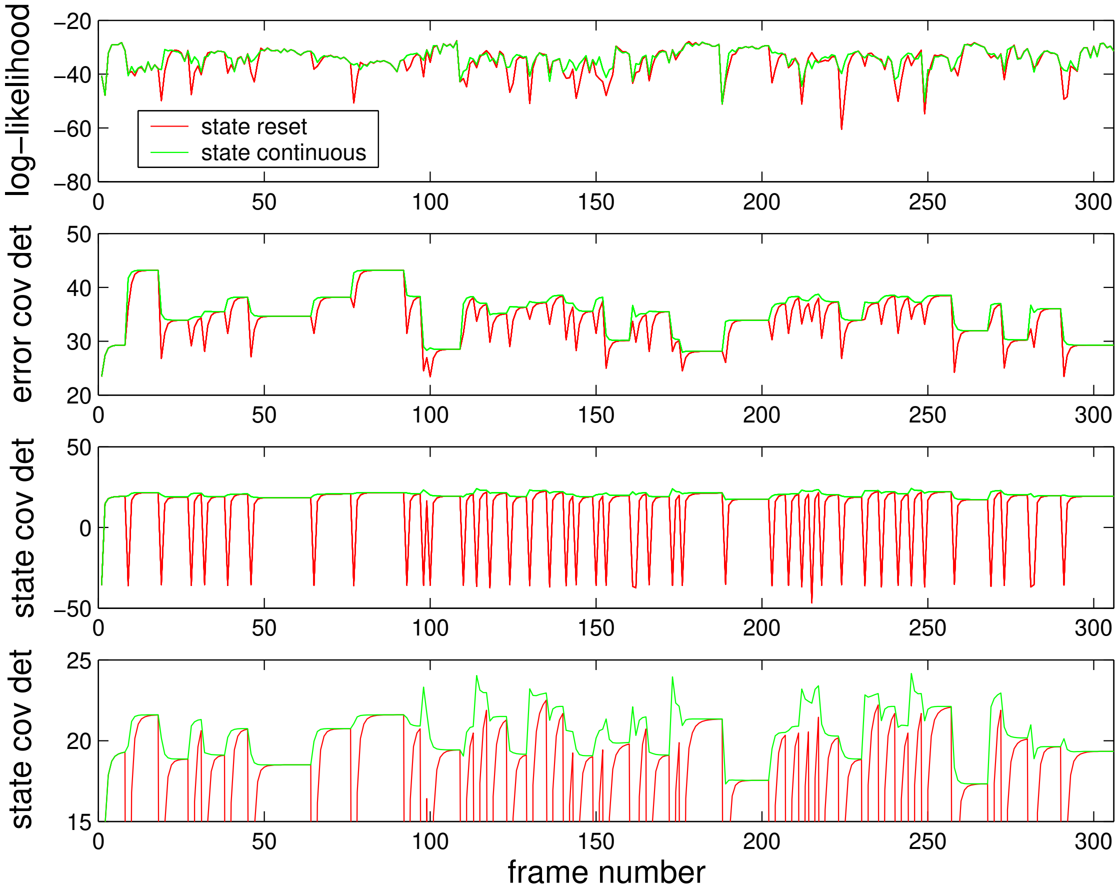

Figure 7.1: Pictorial comparison of the framewise likelihood, error covariance log determinant $\log|\Sigma_{\mathbf{e}_t}|$ and state covariance log determinant $\log|\Sigma_{t|t}|$ with the state reset and passed over boundaries. The 3^{rd} and 4^{th} plots both show $\log|\Sigma_{t|t}|$, the latter with a y-scale which provides greater detail. The plots were produced using an MFCC parameterization of the TIMIT sentence 'even then if she took one step forward he could catch her'.

Similar patterns are apparent in the plot of the error covariance determinant $|\Sigma_{\mathbf{e}_t}|$, though since $\Sigma_{\mathbf{e}_t}$ is a combination of the observation noise covariance C and a projection of the predicted state covariance $H\Sigma_{t|t-1}H^T$, a floor will be provided on its minimum magnitude by C. The spikes at segment boundaries are still evident in the state-reset plot though are less marked. The top plot shows that framewise likelihoods where the state has been passed across phone boundaries are generally equal to or higher than those produced with the state reset. To calculate the framewise likelihood, the prediction errors are normalised by the error covariance and so, as expected, boundary effects are evident in the state-reset likelihood.

It is apparent from these observations that allowing the state to run across model boundaries gives a subtle modification of the properties of the LDM. Such an implementation frequently leads to a slight over-estimation of the state covariance at the start of new segments. In some cases there are sudden increases in its magnitude, such as shown by the spike at frame 100 in the 4^{th} plot of Figure 7.1. Further investigation is required to establish the instances when it is advantageous to pass the state and when resetting is useful. Building an understanding of the manner in which this choice interacts with the ability to make phone class decisions would be non-trivial, though desirable given the intuitive appeal of such a model for ASR.

Decoding with a continuous state

Should continuous state decoding of LDMs be required, there is a practical issue which must be overcome: as observed in Section 6.2 on page 192, the Viterbi criterion can only be applied when the state is reset. Since the state at any time affects future evolution, there is no guarantee that for two paths with identical language model state but differing location in state space $\mathcal{X}$, the path with lower likelihood will not supersede that of the other at some future time. The inadmissibility of the Viterbi criterion would create a vastly increased search space, though it may be that an approximation can be made in which paths with lower likelihood are a removed if the states are 'close'. In fact, the predict-correct nature of the state process means that the influence which the initial conditions have on future state distributions diminishes with each forward Kalman recursion.

To demonstrate this pictorially, a set of reference state mean vectors and covariance matrices was first found by running a forward filter for the [`eh`] and [`sh`] segments from the TIMIT sentence 'Even then if she took one step forward he could catch her.' For the same segments and models, filters were also run whilst varying the state initial distribution by setting it to that of each of the other models in the set. The plots in Figure 7.2 show the root mean square errors (RMSE) between the reference state statistics and those generated with each distinct initial value. The rapid reduction in the RMSE suggests that an approximation to Viterbi could be applied where two or more paths finish at the

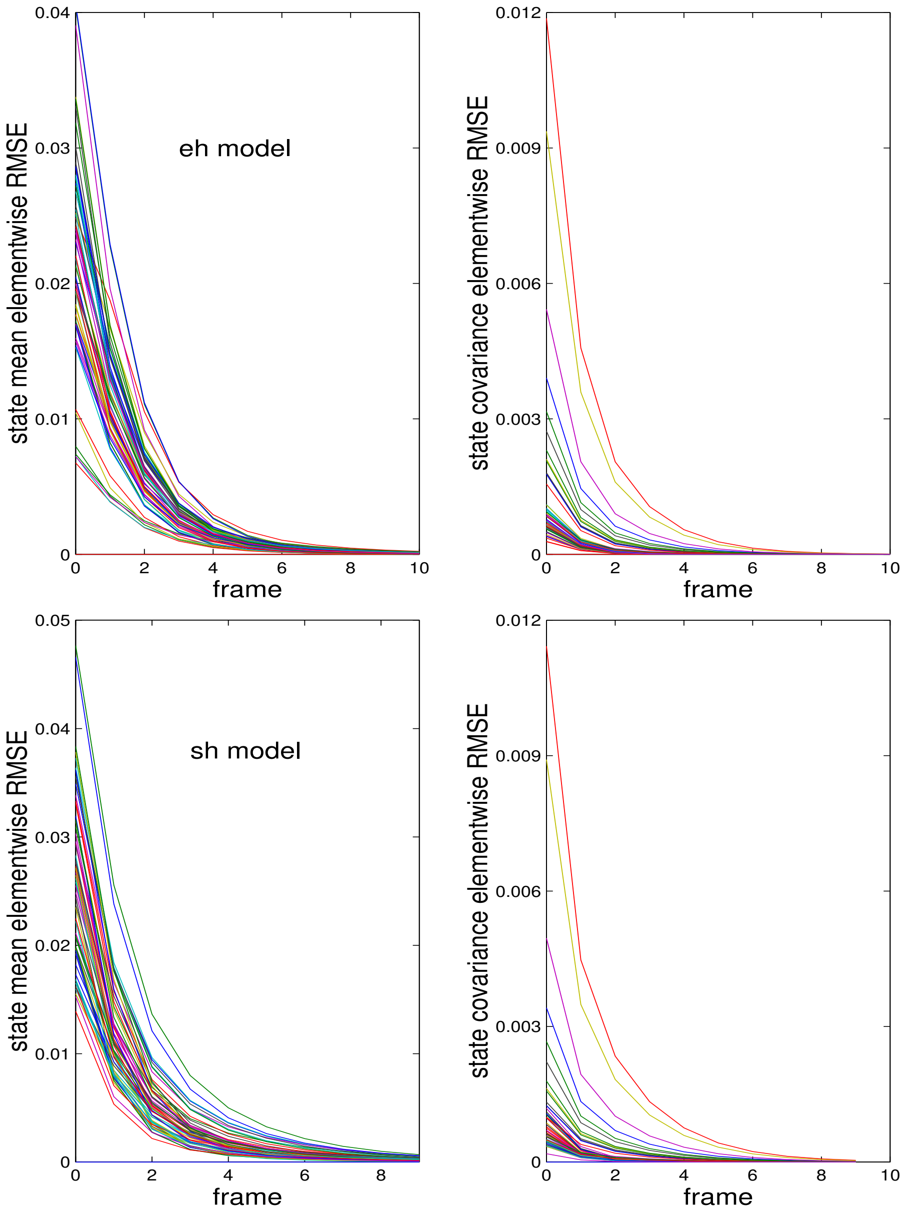

Figure 7.2: A set of reference state mean vectors and covariance matrices was found by running a forward filter for the [eh] segment in 'then' and the [sh] segment in 'she' from the TIMIT sentence 'Even then if she took one step forward he could catch her.' For the same segments and models, filters were also run whilst varying the state initial distribution by setting it to that of each of the other models in the set. The plots show the root mean square errors (RMSE) between the reference state statistics and those generated with each distinct initial value.

same time with identical language model states, and have occupied the same model for a number of frames.

Convergence of filtered quantities

Pre-computation of the 2^{nd} order state statistics and caching of likelihoods is used in state-passed decoding and leads to significant reductions in computation. This is only possible if the initial state distribution is known in advance – which of course relies on resetting at segment boundaries. However, when running a Kalman filter with no inputs, the error covariance $\Sigma_{\mathbf{e}_t}$, Kalman gain K_t, predicted state covariance $\Sigma_{t+1|\,t}$, and corrected state covariance $\Sigma_{t|t}$ as described in Section 4.2.1 on page 89 converge after a few iterations. An *ad hoc* measure of convergence of the state statistics can be made by computing the root mean square error (RMSE) between the entries in successive filtered matrices. Figure 7.3 shows such errors for $\Sigma_{\mathbf{e}_t}$, K_t, $\Sigma_{t+1|\,t}$, and $\Sigma_{t|t}$ for models of each of the 46 MOCHA phones trained on EMA data. Low values indicate small differences between successive matrices. It is apparent that these quantities converge rapidly, with only slight adjustments 4 frames from the start of a segment. Computational savings are thus offered by ceasing to update the error covariance $\Sigma_{\mathbf{e}_t}$, Kalman gain K_t, predicted state covariance $\Sigma_{t+1|\,t}$, and corrected state covariance $\Sigma_{t|t}$ after the fourth or fifth frame of a new model.

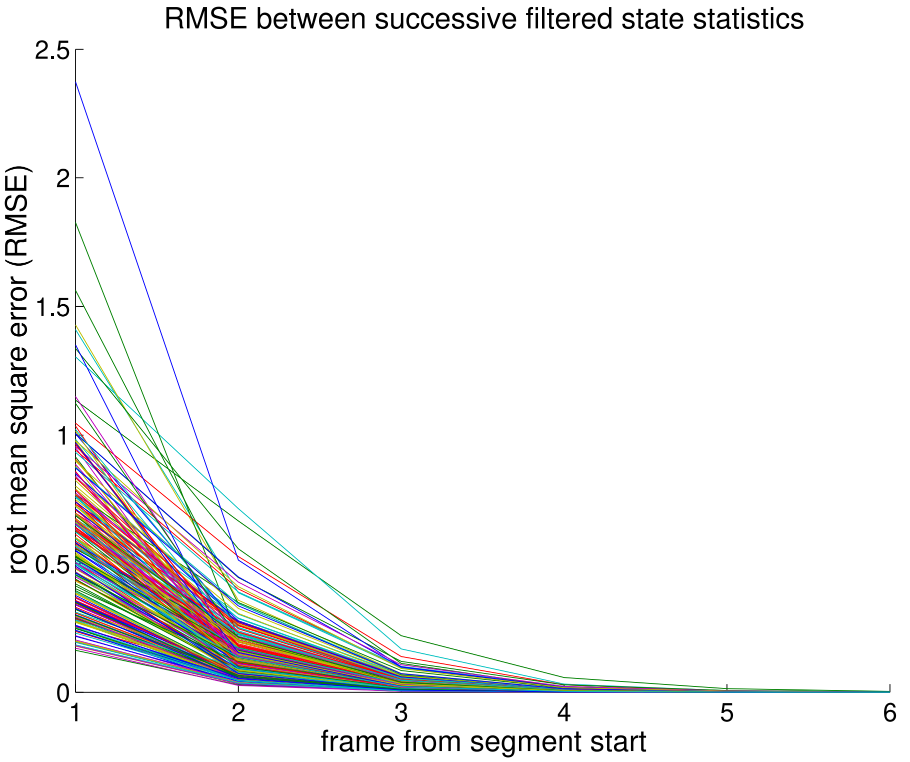

Figure 7.3: A pictorial representation of the convergence of the error covariance $\Sigma_{\mathbf{e}_t}$, Kalman gain K_t, predicted state covariance $\Sigma_{t+1|\,t}$, and corrected state covariance $\Sigma_{t|t}$ during filter updates. Each line shows the diminishing root mean square error (RMSE) between successive values of $\Sigma_{\mathbf{e}_t}$, K_t, $\Sigma_{t+1|t}$, and $\Sigma_{t|t}$ for models of all 46 phones trained on MOCHA EMA data.

7.2 Limitations with the current approach

The approach taken in this thesis has been to work with simple models and tasks which are small enough to allow meaningful analysis of results. In this manner, a deep understanding of the properties of the models in question can be built up: a more complex task, such as conversational speech, whilst being more realistic of the requirements made of an ASR system would also make error analysis significantly more complicated. The following sections outline two of the main limitations with the current system, and how these might be addressed in order to scale up to larger and more demanding tasks.

7.2.1 Training uses phonetic labels

The experiments reported in previous chapters all rely on time-aligned phonetic labels to train the model sets. Very few corpora include manual phone labels, and so systems are typically trained from word transcriptions aligned with the speech signal at the sentence or paragraph level. Full embedded training of LDMs requires integrating over all possible segmentations, which will be computationally expensive (though tractable) as a separate Kalman smoother must be run for each one.

An approximation to full EM training is to alternate between segmenting (a.k.a forced aligning) each utterance using the most recently estimated model parameters and then continuing training. This method uses the single most likely model alignment rather than summing over all possible alignments and is therefore known as Viterbi training. The decoder as described in Section 6.2 can simply be adapted for use as an aligner. At each pop, all hypotheses are removed from the stack apart from the correct one. By storing the candidate phone end-times for each partial hypothesis, a traceback can be made to find the segmentation which produced the final highest likelihood.

A pilot experiment was performed to assess the LDM's ability to provide a segmentation of the data. The highest classification and recognition accuracies on the TIMIT data were found using LDMs with a 9-dimensional state and MFCCs with δ and $\delta\delta$ parameters. The decoder was used with these models to give alignments for all of the utterances from the TIMIT corpus. New labels were prevented from shifting more than 50ms (5 frames) either side of the manual label start and end times. This constraint affected just over

2% of the 194591 labels. An LDM-of-phone classification experiment following the exact same procedure as in Section 5.2.1 on page 149 was carried out. The only variation was that the original set of label files was replaced with the automatically generated ones in both training and testing.

labels	classification accuracy	recognition accuracy
TIMIT manual	72.3%	60.3%
LDM alignments	**75.9%**	59.6%

Table 7.7: Results of TIMIT classification and recognition experiments where phone models have been trained according to a set of LDM alignments rather than manual labels. The LDM alignments were also used in testing during classification.

Results are given in Table 7.7, and show a significant increase in the classification accuracy, from 72.3% to 75.9%. However, using this new model set for recognition gave an accuracy of 59.6%, slightly lower than the equivalent result using models trained on the manual labels which was 60.3%. Whilst Viterbi training did not in this case produce improved recognition performance, the classification result shows that LDMs can successfully be used to align data. This will allow future work to extend to larger corpora for which manual phonetic labels are unavailable.

7.2.2 Unimodal output distributions

The LDM gives a unimodal time-varying Gaussian distribution over the observations. However, the parameterized speech signal consists of data which is only approximately Gaussian (Young 1995). Furthermore, factors such as variability between speakers mean that features can be multimodal. Gaussian mixture models are a frequently-used approach for approximating general probability density functions. Under such a model,

$$\mathcal{Y} \sim \sum_{j=1}^{r} \lambda_j \, p_j(\mathcal{Y}) \tag{7.2}$$

where the p_j are Gaussian distributions $\mathcal{Y}|j \sim N(\theta_j, \phi_j)$ and $\lambda_j = P(j)$ is the prior on mixture component j of r such that $\sum_{i=1}^{r} \lambda_j = 1$ (r is used here rather than the more usual m since models have previously been denoted m). State-of-the-art HMM speech recognition systems such as HTK (Young et al. 2002) use mixtures of Gaussians to model the observations generated by each state. With the addition of mixtures giving substantial performance improvements for HMMs, multimodal modelling for LDMs could be expected to provide similar advantages.

Replacing the Gaussian observation noise distribution $\boldsymbol{\epsilon}_t$ with a mixture distribution would be a simple way in which to produce a multimodal LDM. However, this also leads to a model which is computationally intractable. Inferring the state distribution $\mathbf{x}_t$ would involve conditioning on a mixture density over the observations. Each forward filter recursion would then cause an exponential growth in the number of mixture components with which the state was described. Replacing other model parameters with mixtures results in the same computational intractability. Section 7.3.1 below describes how future work will seek to give a multimodal representation of the observations whilst minimising such effects.

Full covariance observation noise The results of Section 5.2.4 which compared a few variations on the fully parameterized LDM, summarised in Table 7.3 on page 220, showed that a full covariance matrix for the observation noise improved classification accuracy. This requires models with a large number of parameters, and also increases computational expense.

The projection of the state distribution by H determines the approximation which can be made to the error distribution. In this case, the linear mapping does not match the modelling given by estimating a full covariance matrix. A different form of H, such as a general non-linear mapping may improve the ability to capture the correlation structure of the data. This type of approach was taken in Richards & Bridle (1999) and Iso (1993), both of which are outlined in Section 2.3.5 of the literature review. Alternatively, moving toward a multimodal representation of the parameterized speech signal will allow more general output densities and reduce the mismatch between the true and approximated error distributions.

7.3 Future work

Work at CSTR will continue the investigation of linear dynamic models for speech recognition. Future directions are summarised below.

7.3.1 Switching state-space models

The multiple regime (MR) experiments of Section 5.3.2 on page 164 altered the standard implementation by modelling each segment with a series of LDMs. The state process could be continuous for the length of each phone, but the set of LDM parameters used to generate observations was controlled by a deterministic mapping dependent on segment type and duration. The more general case, with the regime changes modelled by a discrete (usually Markovian) process is termed a switching state-space model by Ghahramani & Hinton (1996*b*), and will be referred to as a *switching linear dynamic model* in this work.

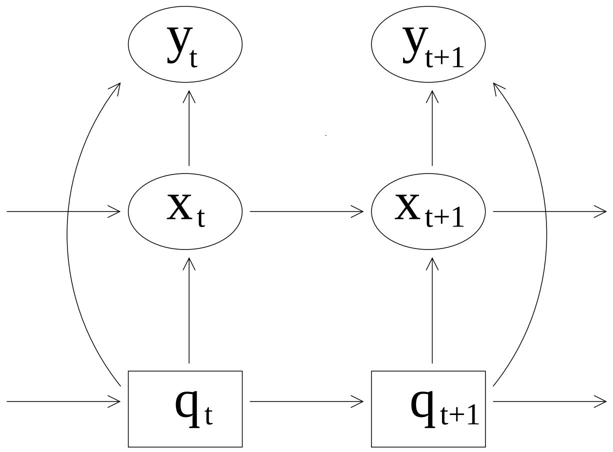

Figure 7.4: Pictorial representation of a switching LDM. Arrows represent dependencies with $\mathbf{x}_t$, $\mathbf{y}_t$ and q_t denoting the state and observation vectors, and discrete switch state at time t respectively.

Figure 7.4 shows such a model represented as a Bayesian network (Smyth 1998). Arrows denote dependencies, with square and oval nodes corresponding to discrete and continuous random variables. As usual, $\mathbf{x}_t$ and $\mathbf{y}_t$ denote the state and observation vectors, and q_t is the discrete switch state at time t.

Depending on the switching topology, this class of models can be used to approximate non-linear dynamics, along with non-Gaussian and multimodal output distributions. The switching process $\mathcal{Q}$ is typically unobserved, and so full inference requires a combination of all possible models $q_j \in \mathcal{Q}$, each weighted by the prior probability of being the true model given the observations up to time t, $P(q_t = j \mid \mathbf{y}_1, \ldots, \mathbf{y}_t)$ (Murphy 1998).

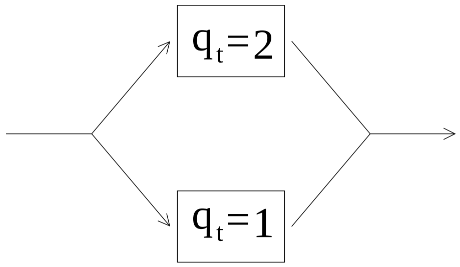

Figure 7.5: Example of a switching topology which allows modelling of non-Gaussian and/or multimodal output distributions.

Figure 7.5 shows a switching topology which would allow modelling of multimodal or non-Gaussian output distributions. If the models associated with switch states 1 and 2 differ in one or more of the observation process parameters H, $\mathbf{v}$ and C, then the output distribution will be a Gaussian mixture model with 2 components. Alternately, temporal switching is shown in Figure 7.6 where the models corresponding to switch states 1 and 2 differ in one or more of their state process parameters F, $\mathbf{w}$ and D. Regime switching of this type gives a piecewise linear state evolution which can be used to approximate non-linear dynamics. The continuous state classification experiments in Section 5.4 on page 176 provide an example of temporal switching where all parameters are switched at known points – in this case phone boundaries.

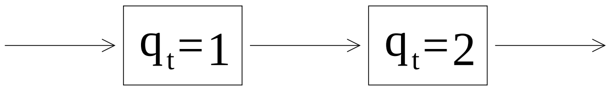

Figure 7.6: Example of a switching topology which can be used to approximate non-linear dynamics.

7.3.2 Tractability

Section 7.2.2 on page 232 above pointed out the downside to including mixtures in LDMs: exponential growth of the number of components in the state distribution leads to computational intractability for all but the simplest of models and shortest of observation sequences. The machine learning literature contains a number of approximate methods which can be used to work around the problems involved in inference and estimation. Possible approaches include:

- N-best Viterbi, where the N most likely components of the state distribution are kept any time.
- Gaussian merging, where the distribution of $\mathbf{x}_t$ is approximated by a Gaussian mixture with a lower number of components.
- Variational methods, in which a tractable distribution is chosen to approximate the true though intractable one. Choosing such a distribution is not always easy in practice, and variational methods can prove computationally expensive.

The switching model described above introduces *mixing* of parameters. In fact this is not a desirable property in an acoustic model for ASR: frame-based HMMs were criticised during the discussion of segmental HMMs on page 33 for being able to generate whilst randomly switching between mixture components at every frame. By regarding switching as an actual *switching* rather than *mixing* process, the benefits of multimodal modelling can be introduced whilst retaining the structure which segment models allow. Since LDMs are used to generate segments and not individual frames, the switching process should only be allowed to change state at certain times, rather than at every frame. Performing N-best Viterbi, or an approximation to Viterbi as described in Section 7.1.4, when the switching process finite-state network paths meet will keep the number of components describing the state small. Computation, which must be considered for a practical ASR system, is significantly reduced by switching rather than mixing.

Once the switching process arrives in a particular state, an associated LDM generates a number of frames using a fixed set of parameters. The switching process then transitions to another state, where another LDM generates another sequence of frames using a new parameter set. Such an approach gives an appealing replication of speech production: at any given time the articulators follow a unimodal path, though context or speaker characteristics dictates the set of trajectories or targets required to produce a given segment.

7.3.3 Automatic topology learning

The application of LDMs to speech data found in this work, along with those of Digalakis (1992), Digalakis & Ostendorf (1992), Digalakis et al. (1993), and Rosti & Gales (2003) forces parameter switches at phone boundaries. Furthermore, the multiple regime LDMs of Section 5.3.2 follow the correlation invariant models of Digalakis (1992) and use deterministic mappings to control the parameter switching within phone models. In all but the pilot studies presented in Section 5.4 and by Rosti & Gales (2003), the states are also reset at the start of each phone, so these are not strictly switching models as defined above. However, the observation that manually forced switch points will be suboptimal is equally applicable.

ASR requires mapping from continuous features to words which are symbolic and discrete. At some level then, the parameterized speech signal must be divided into a series of regimes. However, these regimes are not necessarily neatly abutted like "beads on a string" (Ostendorf 1999), but in fact influence each other strongly. Furthermore, phones are unlikely to the optimal units in all cases, as frequently it may be useful to model shorter regimes such as stop closures or longer ones such as syllables. A project is currently underway at CSTR to automatically derive a unit inventory for speech recognition with LDMs. Building a switching model and learning the topology of the underlying finite-state switching process will accompany this work.

There are no closed-form solutions for the problem of inferring a finite-state model topology from data. The two approaches which appear in the literature either involve state splitting or state merging (Mohri 1997). The first of these initialises with a simple topology and repeatedly splits existing states to add new ones (Ostendorf & Singer 1997, Freitag & McCallum 2000). Alternatively, a complex initial topology can be constructed and then similar states merged or tied (Stolcke & Omohundro 1992, Stolcke & Omohundro 1994, Lee, Kim & Kim 2001). Whichever approach is taken, states are added or removed until convergence has been reached according to some metric.

It may be that either of these methods can be improved using a *prior* on the topology, based on an a combination of interpretation of the role of each model parameter, and application of phonetic and phonological knowledge. As described in Chapter 4, it is the state process which determines the underlying dynamics and the observation process which controls the output distribution. Therefore, switching F can be seen as entering a new regime or segment for which there is a new target in state space $\mathcal{X}$, and switching H can be used to model abrupt changes in the parameterized speech signal such as occurs during the release portion of a plosive. Observations such as these can be used to determine a linguistically motivated *prior* on the state transition network before data driven methods are allowed to take over.

7.4 Final word

In the preamble, the criteria laid down in the oft-cited Bourlard et al. (1996) which must be met to justify risky departures from mainstream approaches to ASR were given. These are:

- solid theoretical or empirical motivations
- sound methodology
- deep understanding of state-of-the-art systems and of the specificity of the new approach.

I hope this thesis demonstrates all of these.

Appendix A

Phone sets

A.1 MOCHA `fsew0` phone set

MOCHA symbol	IPA symbol	example
Vowels – front		
a	æ	h**a**d
e	ɛ	h**ea**d
i	ɪ	h**i**d
ii	iː	h**ee**d
iy	i	eas**y**
Vowels – mid		
@	ə	dodg**er**
@@	ɚ	b**ir**d
uh	ʌ	**u**s
Vowels – back		
aa	ɑ	h**ar**d
o	ɒ	h**o**d
oo	ɔ	h**aw**ed
u	ʊ	g**oo**d
uu	u	p**oo**dle
Diphthongs		
ai	aɪ	hide
ei	eɪ	b**ay**ed
eir	ɛə	h**air**
i@	ɪə	h**ere**
oi	ɔɪ	ah**oy**
ou	oʊ	h**oe**
ow	aʊ	w**o**w

MOCHA symbol	IPA symbol	example
Oral stops – voiced		
b	b	**b**at
d	d	**d**ate
g	ɡ	**g**ate
Oral stops – unvoiced		
p	p	**p**at
t	t	**t**ip
k	k	**c**at
Affricates		
ch	ʧ	**ch**eese
jh	ʤ	**j**ob
Fricatives – unvoiced		
f	f	**f**ell
h	h	**h**orse
s	s	ki**ss**
sh	ʃ	wi**sh**
th	θ	tee**th**
Fricatives – voiced		
v	v	proo**ve**
z	z	la**ze**
zh	ʒ	vi**s**ion
dh	ð	tee**th**e
Liquids		
l	l	**l**eaf
r	ɹ	**r**at
Glides		
w	w	**w**e
y	j	**y**ak
Nasals		
m	m	**m**ake
n	n	**n**ot
ng	ŋ	si**ng**
Other symbols		
breath		breath
sil		silence

A.2 TIMIT phone set

The IPA symbols closest to each TIMITBET phone are taken from Keating (1998).

TIMIT symbol	IPA symbol	example
Vowels – front		
ae	æ	b**at**
eh	ɛ	b**e**t
ih	ɪ	b**i**t
ix	ɨ	deb**i**t
iy	i	b**ee**t
Vowels – mid		
ah	ʌ	b**u**t
ax	ə	**a**bout
ax-h	ə̥	s**u**spect
axr	ɚ	butt**er**
er	ɝ	b**ir**d
Vowels – back		
aa	ɑ	b**o**tt
ao	ɔ	b**ough**t
uh	ʊ	b**oo**k
uw	u	b**oo**t
ux	ʉ	t**oo**t
Diphthongs		
aw	aʊ	b**out**
ay	aɪ	b**i**te
ey	eɪ	b**ai**t
ow	oʊ	b**oa**t
oy	ɔɪ	b**oy**

TIMIT symbol	IPA symbol	example
Oral stops – voiced		
b	b (release only)	**b**ee
d	d (release only)	**d**ay
dx	ɾ	mu**dd**y
g	g (release only)	**g**ay
Oral stops – unvoiced		
k	k (release only)	**k**ey
p	p (release only)	**p**ea
q	ʔ	ba**b**t
t	t (release only)	**t**ea
Oral stop closures		
bcl	b̚	
dcl	d̚	
gcl	g̚	
kcl	k̚	
pcl	p̚	
tcl	t̚	
Affricates		
ch	ʧ(release only)	**ch**eese
jh	ʤ(release only)	**j**ob
Fricatives – unvoiced		
f	f	**f**ell
hh	h	**h**orse
hv	ɦ	a**h**ead
s	s	ki**ss**
sh	ʃ	wi**sh**
th	θ	tee**th**
Fricatives – voiced		
dh	ð	tee**th**
v	v	proo**ve**
z	z	la**ze**
zh	ʒ	vi**si**on

TIMIT symbol	IPA symbol	example
Liquids		
el	l̩	bott**le**
l	l	**l**eaf
r	ɹ	**r**at
Glides		
w	w	**w**e
y	j	**y**ak
Nasals		
em	m̩	botto**m**
en	n̩	butto**n**
eng	ŋ̍	washi**ng**ton
m	m	**m**ake
n	n	**n**ot
ng	ŋ	si**ng**
nx	ɾ̃	wi**nn**er
Other symbols		
epi		epenthetic silence
pau		pause
h#		silence

Appendix B

Model initialisation

Given a model and some data, the EM algorithm updates the model parameters in such a way as to increase the model's likelihood over the data. The iterative nature of EM means that initial estimates must first be found for model parameters. Moreover, whilst EM is guaranteed to increase model likelihood over the data at every iteration, there is the possibility of stepping toward local minima on the error surface. As will be demonstrated, initial conditions can have significant impact on final model performance, though unfortunately there is no one established technique for initialising LDMs.

This appendix presents two possible approaches to parameter initialisation. The first is an *ad hoc* method in which assumptions are made about the function of each parameter, and hence the sorts of values they should take. Starting values for each of the parameters are either then chosen, or found experimentally. The second approach takes the parameters of a factor analyser model to initialises the observation process. These are in turn used to infer state parameters.

It is important that in developing an ASR system, the test set is used as infrequently as possible, to avoid a gradual tuning toward the test data. Therefore, the validation components of the MOCHA and TIMIT classification procedures outlined in Sections 5.1.1 and 5.2.1 on pages 115 and 149 are used to compare combinations of initial conditions. The accuracies quoted below thus refer to the highest classification accuracies found on the validation set for a range of language model scalings and training iterations. As a reminder, the LDM is described by

$$\begin{aligned} \mathbf{y}_t &= H\mathbf{x}_t + \boldsymbol{\epsilon}_t & \boldsymbol{\epsilon}_t &\sim N(\mathbf{v}, C) \\ \mathbf{x}_t &= F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t & \boldsymbol{\eta}_t &\sim N(\mathbf{w}, D) \end{aligned}$$

and a distribution over the initial state, $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$. Observations $\mathbf{y}_t$ and state $\mathbf{x}_t$ vectors have dimensions p and q respectively.

B.1 Ad hoc parameter initialisation

In this approach, assumptions are made about the purpose and interaction of the parameters of the LDM. The initial values are then chosen with the intention of steering model parameters toward these ideals.

The function of a state-space model is to make a distinction between some underlying process and the observations with which it is represented. Ideally, there will be consistent dynamics in the data which can be closely modelled by the state's evolution. In such a situation, predictions would be confident and a low state error covariance would follow. Furthermore, the smoothing required to allow for any mismatch between predictions and observations should be accounted for by the observation noise. Capturing the correlation structure of the data in the observation matrix H would be desirable as the components of the state would be independent. This would allow for an efficient implementation of the model as the initial state covariance Λ, state error covariance D and state evolution matrix F would all be diagonal.

B.1.1 Method

To encourage the model to function as above, rather than producing an uninformative state process and modelling the data as observation noise, the state error is initialised to be small compared to the observation noise. The covariance matrices C, D and Λ are set to be diagonal, with a ratio of 1:5 chosen for the relative magnitudes of the non-zero elements of D and C. The state process noise mean, $\mathbf{w}$, and the initial state mean, $\mathbf{x}_1$, are both set to $\mathbf{0}_q$, a q-dimensional zero vector. The observation noise mean, $\mathbf{v}$, is initialised as the overall mean of the data.

F is set to be an identity matrix, meaning that the evolution of the state space is simply a random walk. Note that after one iteration of EM, the constraint of F being a decaying mapping as described in Section 4.2.4 is applied, though remedial action is rarely necessary. H is either set to be uniform across the the dimensions of the state so that every entry is fixed as $1/q$, or randomised with each element drawn independently from a $N(0, 0.5)$ distribution.

B.1.2 Results

MOCHA EMA data

For both H randomised and given fixed equal values across state dimensions, F, $\mathbf{w}$, $\mathbf{v}$ and $\mathbf{x}_1$ were set as above. The state initial covariance was initialised as I_q, a q-dimensional identity matrix. Table B.1 shows the classification validation accuracies for the observation noise C set to γI_p and the state noise D as $\frac{\gamma}{5} I_q$ where γ varies from 0.0005 up to 5000, each time by an order of magnitude. The best accuracy of 59.7% is given with

γ	0.0005	0.005	0.05	0.5	5	50	500	5000
H fixed	38.9%	46.5%	57.1%	57.8%	57.2%	57.2%	57.0%	56.9%
H randomised	28.2%	45.0%	56.6%	59.4%	**59.7%**	59.6%	59.1%	58.0%

Table B.1: Classification validation accuracies for systems trained on MOCHA EMA data for a variety of magnitudes of the noise covariances. C is initialised as γI_p and D as $(\gamma/5)I_q$. H is either given equal weightings for each of the dimensions of the state vector or randomised

H randomised and $\gamma = 5$, meaning that $C = 5I_p$ and $D = I_q$. These values are kept and accuracies obtained for the initial state covariance, Λ, set to ξI_q with ξ ranging from

ξ	0.0001	0.001	0.01	0.1	1	10	100	1000
H randomised	59.5%	60.2%	**60.7%**	59.2%	59.7%	59.0%	60.1%	60.1%

Table B.2: Classification validation accuracies for systems trained on MOCHA EMA data for a variety of magnitudes of state initial covariance, Λ, set to ξI_q.

0.0001 to 1000, again by an order of magnitude each time. The results are given in Table B.2. The best validation accuracy of 60.7% is given for $\Lambda = 0.01 I_q$.

Now, having found some initial parameter estimates in which all models are initialised identically, phone-specific initial observation noise means are considered. These are calculated by averaging the observation vectors corresponding to each phone type over the training data. The results in Table B.3 show that a universal data mean gives higher clas-

features	overall mean	phone-specific mean
EMA	**60.7%**	59.3%

Table B.3: Taking the initial parameter estimates chosen above, initialising with phone-specific observation noise means is compared to using one universal data mean.

sification accuracies on the validation set. Phone specific means not only give a higher log-likelihood on the training data, 11.6 compared to 7.6, but also on the validation data, 8.7 compared to 5.9. It seems that model specific means improve the fit of the models to each class, but not the ability to discriminate between them.

TIMIT PLP and MFCC data

The same process was carried out for both PLP and MFCC features for speaker-independent classification on the TIMIT corpus. Table B.4 shows that with the state noise covariance

features	γ	0.0005	0.005	0.05	0.5	5	50	500	5000
PLP	H fixed	57.9%	58.8%	59.3%	59.3%	59.2%	59.1%	59.1%	-
	H randomised	28.1%	45.6%	59.4%	**60.9%**	60.8%	60.2%	59.8%	-
MFCC	H fixed	59.2%	59.2%	59.2%	59.0%	60.9%	60.6%	60.3%	-
	H randomised	28.1%	28.4%	29.0%	33.7%	60.2%	61.2%	**61.4%**	60.3%

Table B.4: Classification validation accuracies for systems trained on TIMIT PLP and TIMIT MFCC data for a variety of magnitudes of the noise covariances. C is initialised as γI_p, D as $0.2\gamma I_q$, and H is either initialised to have equal weightings for each of the dimensions of the state vector, or randomised

set to be the identity matrix I, the highest classification accuracies were for γ of 0.5 and 500 for PLP and MFCC features respectively. In both cases the highest accuracies were given for H randomised. Table B.5 shows the effect on the validation accuracy of varying the initial state covariance estimate whilst using the values for C and D chosen

ξ	0.0001	0.001	0.01	0.1	1	10	100	1000
PLP	61.6%	61.7%	**62.0%**	60.1%	60.9%	60.9%	60.9%	60.9%
MFCC	60.7%	60.7%	60.7%	60.7%	**61.4%**	60.7%	60.9%	60.6%

Table B.5: Classification validation accuracies for systems trained on TIMIT PLP and TIMIT MFCC data for a variety of magnitudes of the state initial covariance, Λ, which was set to ξI_q.

above. The highest accuracy was given for Λ of $0.01 I_q$ and I_q for PLP and MFCC features respectively. Table B.6 shows that, as in the case of the MOCHA EMA data, using a

features	overall mean	phone-specific mean
PLP	**62.0 %**	60.9%
MFCC	**61.4%**	60.7%

Table B.6: Taking the initial parameter estimates chosen above, phone-specific observation noise means are compared to one universal data mean for TIMIT MFCC and PLP data.

single data mean to initialise the observation noise mean, $\mathbf{v}$, gives a better accuracy than using phone specific initial means.

B.2 Factor Analysis model for initialisation

Another, possibly more principled, approach to parameter initialisation for LDMs uses a factor analysis model to provide the LDM observation process parameters. A factor analyser (see Section 2.1.2 on page 19) can be cast as an LDM observation process conditioned on a static standard Gaussian state target, and so consists only of H, $\mathbf{v}$ and C. EM can be used to estimate these parameters, and setting $\mathbf{v}$ to be the data mean, only initial values for H and C need to be chosen.

Once a factor analyser model has been trained for each phone, H, $\mathbf{v}$ and C are used directly as the LDM observation process, and are then used to provide estimates of the remaining parameters following some the ideas in Section 4.2.2 on page 93.

B.3 Method

With the observation process parameters fixed, the LDM joint log likelihood of state and observations given in Equation 4.19 on page 93 becomes:

$$\begin{aligned} l(\Theta|\mathcal{Y},\mathcal{X}) &\propto -\sum_{t=1}^{N}\left\{log|D| + (\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})^T D^{-1}(\mathbf{x}_t - F\mathbf{x}_{t-1} - \mathbf{w})\right\} \\ &\quad - \log|\Lambda| + (\mathbf{x}_1 - \boldsymbol{\pi})^T\Lambda^{-1}(\mathbf{x}_1 - \boldsymbol{\pi}) \end{aligned} \tag{B.1}$$

which can be maximised for each parameter in turn to give

$$\begin{aligned} \begin{bmatrix} \hat{F} & \hat{\mathbf{w}} \end{bmatrix} &= \begin{bmatrix} \sum_{t=1}^n \mathbf{x}_t\mathbf{x}_{t-1}^T & \sum_{t=1}^n \mathbf{x}_t \end{bmatrix} \begin{bmatrix} \sum_{t=1}^n \mathbf{x}_{t-1}\mathbf{x}_{t-1}^T & \sum_{t=1}^n \mathbf{x}_{t-1} \\ \sum_{t=1}^n \mathbf{x}_{t-1}^T & 1 \end{bmatrix}^{-1} \\ \hat{D} &= \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}_t\mathbf{x}_t^T - \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}_t\mathbf{x}_{t-1}^T F^T - \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}_t\mathbf{w}^T \\ \hat{\boldsymbol{\pi}} &= \mathbf{x}_1 \\ \hat{\Lambda} &= \mathbf{x}_1\mathbf{x}_1^T - \mathbf{x}_1\boldsymbol{\pi}^T \end{aligned}$$

Since $\mathbf{x}_t$, $\mathbf{x}_{t-1}\mathbf{x}_{t-1}^T$ and $\mathbf{x}_t\mathbf{x}_{t-1}^T$ are unknown, they are replaced using their posterior estimates under the original factor analysis model which was used to generate H, $\mathbf{v}$ and C. These expectations are computed as:

$$E[\mathbf{x}_t|\mathcal{Y},\Theta^g] = \hat{\mathbf{x}}_{t|N} \tag{B.2}$$

$$E[\mathbf{x}_t\mathbf{x}_t^T|\mathcal{Y},\Theta^g] = \Sigma_{t|N} + \hat{\mathbf{x}}_{t|N}\hat{\mathbf{x}}_{t|N}^T \tag{B.3}$$

$$E[\mathbf{x}_t\mathbf{x}_{t-1}^T|\mathcal{Y},\Theta^g] = \hat{\mathbf{x}}_{t|N}\hat{\mathbf{x}}_{t-1|N}^T \tag{B.4}$$

Note that Equation B.4 differs from Equation 4.36 which computes a similar quantity under an LDM. The static nature of the factor analysis model means that the state cross-covariance is zero. The state posterior distribution under a factor analysis model which is required to compute these estimates is given in Equation 2.15 on page 19.

B.3.1 Results

EMA data

Classification validation results on MOCHA EMA data using LDMs which were initialised using factor analyser models are shown in Table B.7. The number of EM iterations used in training the factor analyser ranged from 1 to 5, and final classification performance on the validation set deteriorated with each extra iteration. The highest accuracy of 57.8% was obtained using a factor analyser trained for a single iteration, and was lower than the

FA model training iterations	1	2	3	4	5
classification accuracy	**57.8%**	57.6%	57.0%	55.8%	53.8%

Table B.7: Classification accuracies on the validation set after LDMs were initialised using factor analyser models. Results are shown for 1 through to 5 iterations of EM on the factor analyser before LDM training began. Classification was speaker-dependent and used the MOCHA EMA data.

60.7% gained using the *ad hoc* method above.

TIMIT PLP and MFCC data

Similar results were obtained with TIMIT acoustic features, shown in Table B.8. Here, the highest accuracies of 53.1% and 59.9% for PLP and MFCC features respectively are both lower than the accuracies of 62.0% and 61.4% gained initialising with the *ad hoc* method above. The largest deterioration in performance from using this approach to

features	FA model training iterations	1	2	3	4	5
PLP	classification accuracy	**53.1%**	47.3%	45.1%	44.8%	43.7%
MFCC	classification accuracy	**59.9%**	58.8%	57.4%	56.1%	55.0%

Table B.8: Classification accuracies on the validation set after LDMs were initialised using factor analyser models. Results are shown for 1 through to 5 iterations of EM on the factor analyser before LDM training began. Classification was speaker-independent and used the TIMIT MFCC and PLP data.

model initialisation occurs for the PLP features. This is despite the log-likelihood of the models on the validation set increasing from 11.8 to 14.4. As with phone-specific initial means, the fit of the model but not the discriminatory power has been improved.

B.4 Conclusions

The degeneracy present in the LDM (discussed in Section 2.1.2 on page 20) coupled with the iterative, non-optimal nature of EM training means that well chosen initial conditions are important for successful application of LDMs. This is demonstrated in the results above. For example, the highest and lowest accuracies using TIMIT PLP data were 62.0% and 43.7% – significant variation given seemingly sensible initial conditions. In an extreme case, estimation could lead to an H filled with zeros, and all modelling through the observation noise. This would however be preferable to a poorly fitting state process with associated high error covariances which can be an effect of badly chosen initial conditions.

In this appendix, two methods of finding initial estimates for LDM parameters were compared. The first used a combination of hand-picked and experimentally chosen values. In the second, a factor analysis model was used to find estimates of the observation process parameters, and these in turn were used to derive the remaining parameters. For each of MOCHA EMA, TIMIT MFCC and TIMIT PLP features, the *ad hoc* method gave higher classification accuracies on the validation sets, and was adopted for experimentation in this thesis.

Appendix C

TIMIT validation speakers

dialect region	speakers
dr1	mcpm0 mpgh0 mtpf0 fkfb0 fdml0
dr2	mdlb0 mdwd0 mjde0 mjpm0 mkjo0 mmxs0 mrfk0 faem0 fdas1 fjkl0
dr3	mddc0 mdlh0 mfmc0 mjda0 mkls1 mmeb0 mrds0 fgcs0 fljd0 fsjs0
dr4	mjac0 mjls0 mjws0 mljc0 mmgc0 mrab1 msmc0 fkdw0 fsak0
dr5	mdas0 mewm0 mhmg0 mjwg0 mmvp0 mrew1 msas0 fkkh0 fmpg0 fskp0
dr6	mrxb0 msjk0 mtju0 frjb0 fsgf0
dr7	mbth0 mded0 mdlr1 mgar0 mjai0 mkag0 mntw0 mrmg0 fleh0 fpac0
dr8	mmea0 mmpm0 fklh0

Table C.1: Validation speakers used training models on the TIMIT corpus. The distribution of the dialect regions and genders approximates that in the test set.

Appendix D

Tools used in experimental work

The core experimental work reported in this thesis has required writing a library of functions dealing with the implementation of linear Gaussian models. A variety of tools produced by others have also been used in tasks such as feature extraction, language modelling and Viterbi search. The main elements are listed below, with italic used where the code has been written as part of the work for this thesis. An LDM toolkit built on the Edinburgh Speech Tools library is planned for future release.

D.1 General

- Acoustic feature extraction - HTK version 3.1 (Young et al. 2002)
- Numerical differentiation - Edinburgh Speech Tools (Taylor et al. 1997-2003)
- *Linear discriminant analysis* - implemented in `Matlab`, based on a tutorial paper by Balakrishnama & Ganapathiraju (1998)
- Gamma distribution estimation - code by Laurence Malloy
- Viterbi decoding - Edinburgh Speech Tools (Taylor et al. 1997-2003)
- Linear regression and alternating conditional expectation (ACE) algorithm (Breiman & Friedman 1985) - standard `S+` library

- Language modelling - Edinburgh Speech Tools (Taylor et al. 1997-2003) and CMU-Cambridge Statistical Language Modelling toolkit (Clarkson & Rosenfeld 1997)

D.2 Acoustic modelling

The following are all implemented in `C++`, and use the base classes provided by the Edinburgh Speech Tools library (Taylor et al. 1997-2003).

- Factor analyser
  - *Parameter estimation*
  - *Classification routine*
  - *input/output*
- Linear dynamic model
  - *LDM class* (reference-counting for parameter tying by Simon King)
  - *Parameter estimation*
  - *Continuous state parameter estimation*
  - *Classification routine*
  - *input/output*

D.3 Decoding

Decoding of LDMs is implemented within the framework of a flexible, modular stack decoder originally written in `C++` by Simon King at CSTR, again using the base classes provided by the Edinburgh Speech Tools library (Taylor et al. 1997-2003).

- stack and associated operations - pushing, popping, ordering, beam pruning by Simon King.
- *LDM acoustic matching and adaptation of grid routines for segment models*
- *Adaptive pruning*
- *Modification of decoder for continuous-state classification*

Appendix E

Full classification results

The following tables give the classification results of chapter 5 in full. The highest accuracy in each table is given in bold face. In the case of MOCHA data, these are the state dimensions which were then used to produce the cross-validation classification results. Where network-recovered data has been used or compared with other results, K-fold cross-validation was not possible and † marks the result which gave the highest performance on a separate validation set and is taken as best for that particular feature. Where TIMIT data has been used, the highest accuracy is marked in bold face, and † marks the result which corresponds to the best performance on validation data and was therefore taken to be the final result for the features and models in question.

MOCHA EMA factor analyser results			
dimension	classification accuracy		
	EMA	EMA + δ	EMA + δ + $\delta\delta$
1	51.1%	55.1%	56.8%
2	54.0%	56.1%	56.8%
3	55.1%	58.4%	58.7%
4	54.8%	59.4%	61.0%
5	55.1%	60.9%	61.5%
6	56.7%	61.9%	62.6%
7	55.6%	62.8%	62.2%
8	56.7%	63.3%	62.5%
9	57.0%	63.1%	62.6%
10	54.5%	63.1%	61.2%
11	57.0%	**64.1%**	62.2%
12	**57.2%**	62.6%	61.2%
13	56.9%	63.3%	62.3%
14	57.1%	63.4%	62.2%
15	56.9%	63.6%	63.2%
16	56.9%	63.8%	**63.5%**
17	56.9%	63.4%	63.2%
18	56.9%	63.4%	63.0%
19	57.1%	63.6%	62.9%
20	56.8%	62.8%	63.3%

Table E.1: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were EMA data, EMA data with δ coefficients, and EMA data with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

MOCHA EMA LDM results			
dimension	classification accuracy		
	EMA	EMA $+ \delta$	EMA $+ \delta + \delta\delta$
0	57.0%	64.0%	65.4%
1	57.9%	65.0%	64.7%
2	57.8%	64.8%	65.4%
3	58.8%	63.6%	64.6%
4	57.6%	65.7%	65.4%
5	59.7%	65.2%	65.9%
6	60.2%	65.1%	66.0%
7	59.2%	64.2%	66.0%
8	60.7%	64.8%	65.4%
9	60.8%	65.3%	65.4%
10	59.2%	65.0%	65.9%
11	59.5%	64.8%	66.5%
12	60.0%	65.6%	64.6%
13	**60.9%**	63.3%	65.9%
14	59.5%†	64.5%	65.0%
15	60.7%	66.4%	66.7%
16	59.0%	67.0%†	66.7%
17	60.3%	66.8%	**68.3%**
18	60.8%	66.6%	67.8%†
19	59.4%	**67.6%**	67.5%
20	59.4%	66.0%	66.6%
21	59.5%	67.0%	67.8%
22	57.2%	65.2%	67.6%

Table E.2: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were EMA data, EMA data with δ coefficients, and EMA data with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22. These results are shown graphically in figure 5.3 on page 123

MOCHA extended articulatory factor analyser results				
dimension	classification accuracy			
	all artic	all artic + δ	all artic + δ + $\delta\delta$	LDA all artic + δ + $\delta\delta$
1	60.4%	64.0%	63.9%	65.4%
2	62.1%	64.5%	64.6%	69.0%
3	61.3%	65.5%	66.5%	69.7%
4	64.0%	66.4%	66.8%	69.3%
5	64.9%	67.1%	69.6%	69.5%
6	65.3%	69.4%	68.0%	70.4%
7	**66.5%**	69.6%	68.6%	70.5%
8	66.5%	69.2%	68.9%	70.0%
9	64.8%	70.3%	70.4%	70.0%
10	65.5%	69.0%	69.6%	69.8%
11	65.0%	69.1%	69.7%	71.4%
12	66.4%	69.7%	70.1%	70.3%
13	66.3%	70.6%	70.1%	71.8%
14	64.9%	70.5%	70.1%	72.1%
15	65.2%	70.8%	**71.7%**	70.4%
16	64.8%	69.1%	70.9%	72.2%
17	65.0%	70.6%	70.3%	**72.3%**
18	65.4%	71.0%	70.5%	71.4%
19	65.0%	70.4%	70.5%	71.9%
20	65.3%	70.9%	70.9%	72.1%
21	65.3%	**71.4%**	71.2%	72.0%
22	65.1%	70.4%	71.2%	71.7%
23	65.0%	71.1%	70.4%	72.2%
24	65.0%	71.2%	70.5%	72.0%

Table E.3: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were the full articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies are shown for a state dimensions of 0 to 24 and with the data used raw, or post-processed using LDA.

MOCHA extended articulatory diagonal covariance LDM results				
dimension	classification accuracy			
	all artic	all artic $+\delta$	all artic $+\delta+\delta\delta$	LDA all artic $+\delta+\delta\delta$
0	59.9%	62.7%	63.3%	64.9%
1	59.4%	61.8%	64.0%	66.1%
2	62.1%	64.9%	64.4%	67.8%
3	64.2%	65.9%	68.6%	69.4%
4	62.1%	68.0%	68.8%	69.8%
5	64.7%	67.6%	68.1%	71.2%
6	62.3%	68.4%	69.0%	71.0%
7	61.7%	66.8%	69.1%	70.4%
8	64.1%	68.3%	69.0%	71.8%
9	64.8%	67.9%	**70.0%**	71.8%
10	63.4%	69.1%	67.0%	71.5%
11	63.3%	68.9%	67.5%	71.6%
12	**65.1%**	68.9%	68.7%	71.5%
13	64.7%	68.0%	68.1%	71.7%
14	64.5%	69.1%	69.0%	71.2%
15	63.8%	68.6%	67.6%	70.5%
16	64.3%	68.4%	68.9%	71.7%
17	62.5%	68.8%	69.2%	**72.3%**
18	63.1%	68.9%	69.6%	71.4%
19	63.5%	**69.3%**	67.9%	71.4%
20	62.6%	69.3%	68.6%	71.1%
21	64.2%	68.1%	68.9%	71.2%
22	62.1%	68.6%	67.8%	71.6%
23	63.4%	68.6%	69.6%	71.9%
24	61.7%	69.1%	69.2%	71.2%

Table E.4: Speaker-dependent classification accuracies for systems with diagonal covariance LDMs used as the acoustic model. The features were the full articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies are shown for a state dimensions of 0 to 24 and with the data used raw, or post-processed using LDA.

MOCHA extended articulatory diagonal state covariance LDM results				
dimension	classification accuracy			
	all artic	all artic + δ	all artic + δ + $\delta\delta$	LDA all artic + δ + $\delta\delta$
1	66.5%	71.1%	70.8%	71.7%
2	67.6%	71.4%	71.3%	72.4%
3	67.2%	71.4%	71.2%	72.2%
4	67.4%	71.4%	72.1%	72.8%
5	66.7%	72.2%	72.3%	72.8%
6	67.6%	72.4%	72.5%	73.1%
7	66.2%	72.4%	72.1%	73.0%
8	67.4%	72.2%	73.2%	73.4%
9	66.9%	72.4%	72.5%	73.2%
10	67.5%	72.2%	72.6%	73.2%
11	67.7%	72.1%	73.2%	73.1%
12	68.7%	73.0%	73.8%	**75.0%**
13	68.6%	71.9%	72.5%	74.3%
14	68.8%	73.1%	74.2%	72.8%
15	**69.3%**	73.1%	73.1%	72.7%
16	68.7%	72.9%	73.7%	74.0%
17	67.2%	73.2%	72.6%	73.9%
18	69.0%	73.8%	72.8%	74.4%
19	67.9%	74.3%	73.3%	74.4%
20	68.1%	73.4%	72.7%	73.6%
21	68.7%	74.0%	**74.5%**	73.5%
22	66.8%	**74.8%**	73.6%	74.0%
23	69.0%	72.8%	73.1%	73.7%
24	68.1%	74.0%	73.7%	73.2%

Table E.5: Speaker-dependent classification accuracies for systems with diagonal state covariance LDMs used as the acoustic model. The features were the full articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies are shown for a state dimensions of 0 to 24 and with the data used raw, or post-processed using LDA.

MOCHA extended articulatory state covariance identity LDM results				
dimension	classification accuracy			
	all artic	all artic $+\delta$	all artic $+\delta+\delta\delta$	LDA all artic $+\delta+\delta\delta$
1	65.8%	72.0%	71.5%	72.0%
2	66.5%	70.9%	71.1%	72.3%
3	67.0%	70.6%	71.5%	72.4%
4	67.9%	71.1%	71.8%	72.6%
5	66.2%	71.3%	71.8%	72.7%
6	67.0%	71.7%	72.3%	72.4%
7	67.3%	72.3%	72.6%	73.1%
8	67.5%	71.3%	72.5%	73.2%
9	68.4%	72.4%	72.8%	73.1%
10	67.7%	72.4%	73.4%	72.2%
11	67.8%	72.1%	73.2%	73.9%
12	68.4%	73.4%	73.2%	73.6%
13	69.0%	73.0%	73.1%	73.2%
14	68.3%	72.6%	73.3%	71.8%
15	69.5%	73.4%	73.7%	72.8%
16	**70.3%**	72.4%	72.9%	73.2%
17	67.8%	72.6%	74.1%	73.1%
18	69.0%	72.5%	73.7%	**74.6%**
19	68.3%	**74.1%**	74.0%	74.2%
20	67.8%	72.8%	74.2%	73.5%
21	68.2%	73.3%	73.3%	73.2%
22	69.2%	72.6%	**74.5%**	73.0%
23	68.6%	72.8%	73.8%	73.8%
24	69.2%	73.5%	73.9%	73.3%

Table E.6: Speaker-dependent classification accuracies for systems with identity state covariance LDMs used as the acoustic model. The features were the full articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies are shown for a state dimensions of 0 to 24 and with the data used raw, or post-processed using LDA.

MOCHA extended articulatory LDM results				
dimension	classification accuracy			
	all artic	all artic + δ	all artic + δ + $\delta\delta$	LDA all artic + δ + $\delta\delta$
0	65.7%	70.7%	71.0%	71.5%
1	66.2%	71.1%	71.2%	71.9%
2	65.6%	70.8%	71.2%	71.7%
3	66.2%	71.1%	71.8%	72.8%
4	67.0%	71.5%	72.2%	73.0%
5	67.6%	72.3%	72.5%	73.1%
6	67.8%	72.2%	71.8%	73.6%
7	68.1%	70.7%	72.9%	73.7%
8	68.3%	72.2%	71.8%	73.1%
9	68.3%	72.4%	72.4%	73.2%
10	68.3%	72.0%	72.9%	72.7%
11	68.1%	72.3%	73.0%	73.2%
12	69.1%	72.5%	72.5%	73.9%
13	69.1%	73.3%	72.5%	73.3%
14	69.1%	72.6%	72.9%	72.3%
15	69.3%	73.3%	**74.0%**	71.8%
16	68.4%	72.8%	73.6%	72.8%
17	68.6%	71.9%	73.2%	73.1%
18	68.0%	72.8%	73.4%	73.4%
19	68.6%	**73.7%**	73.4%	72.9%
20	68.6%	73.5%	71.9%	71.9%
21	**69.5%**	72.9%	74.0%	72.9%
22	68.7%	73.1%	73.6%	**74.5%**
23	69.0%	72.9%	72.9%	74.0%
24	69.1%	72.8%	73.1%	71.7%

Table E.7: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were the full articulatory set from the MOCHA corpus comprising EMA, laryngograph and EPG data. Accuracies are shown for a state dimensions of 0 to 24 and with the data used raw, or post-processed using LDA. These results are shown graphically in figure 5.4 on page 125.

MOCHA articulatory features from Wrench (2001) LDM results	
state dimension	classification accuracy
	EMA + LAR + $\delta + \delta\delta$ LDA
0	72.9%
1	72.7%
2	72.4%
3	74.2%
4	72.7%
5	73.7%
6	73.6%
7	73.4%
8	73.9%
9	**74.5%**†
10	73.8%
11	73.1%
12	73.7%
13	73.6%
14	74.2%
15	73.9%
16	73.5%

Table E.8: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were an articulatory set derived from the MOCHA corpus and used in Wrench (2001). Accuracies are shown for a state dimensions of 0 to 16. These features described in section 6.3.1.

MOCHA network-recovered EMA factor analyser results			
dimension	classification accuracy		
	NET	NET + δ	NET + δ + $\delta\delta$
1	45.1%	39.5%	45.5%
2	45.6%	45.6%	45.8%
3	47.1%	45.7%	47.5%
4	48.0%	47.3%	48.1%
5	48.6%	47.4%	47.4%
6	48.4%	48.1%	49.1%
7	48.6%	48.0%	48.4%
8	48.9%	48.7%	49.5%
9	47.7%	49.3%	49.8%
10	48.9%	**49.8%**	49.4%
11	49.2%	49.2%	49.5%
12	49.0%	48.8%	50.6%
13	48.8%	48.7%	50.4%
14	49.7%	48.6%	50.2%
15	49.2%	49.2%	50.4%
16	49.4%†	48.5%	50.0%
17	49.9%	49.1%	50.3%
18	49.7%	49.2%	**51.2%**
19	**50.0%**	49.4%	50.5%†
20	49.5%	49.5%†	51.1%
21	49.5%	49.4%	50.3%
22	49.4%	49.3%	50.7%

Table E.9: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were net EMA data, net EMA data with δ coefficients, and net EMA data with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22.

MOCHA network-recovered EMA LDM results			
dimension	classification accuracy		
	NET	NET + δ	NET + δ + $\delta\delta$
0	55.1%	58.2%	59.4%
1	55.7%	58.2%	59.1%
2	57.0%	58.2%	59.1%
3	56.6%	59.7%	59.0%
4	56.4%	60.0%	59.0%
5	57.2%	59.7%	59.1%
6	57.5%	**60.5%**	59.0%
7	56.8%	58.2%	58.3%
8	57.0%	59.1%	59.1%
9	57.1%†	59.7%	59.0%
10	56.9%	59.6%	59.4%
11	56.7%	59.4%	59.2%
12	57.5%	59.8%	59.0%
13	**57.7%**	59.3%†	59.1%
14	57.0%	59.2%	58.7%
15	55.9%	59.2%	59.4%
16	57.0%	59.8%	**59.6%**
17	57.0%	59.8%	59.1%
18	56.2%	59.5%	59.2%
19	56.9%	58.9%	59.2%
20	57.6%	60.4%	59.6%†
21	55.9%	59.6%	58.9%
22	55.7%	60.1%	59.0%

Table E.10: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were net EMA data, net EMA data with δ coefficients, and net EMA data with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22. These results are shown graphically in figure 5.6 on page 129.

MOCHA PLP factor analyser results			
dimension	classification accuracy		
	PLP	PLP $+\delta$	PLP $+\delta+\delta\delta$
1	64.6%	66.2%	62.9%
2	68.1%	66.7%	63.3%
3	68.5%	68.2%	66.0%
4	67.6%	69.7%	66.4%
5	68.6%	70.1%	67.6%
6	69.5%	71.5%	67.1%
7	70.0%	72.5%	67.6%
8	69.5%	72.7%	67.8%
9	70.1%	72.0%	69.3%
10	69.8%	72.2%	69.0%
11	70.3%	71.6%	69.5%
12	69.6%	72.3%	69.3%
13	70.3%	**72.9%**	69.6%
14	69.9%	71.6%	70.0%
15	70.3%	72.3%	70.7%
16	70.2%	72.8%	70.6%
17	**70.4%**	72.8%	70.1%
18	70.0%	72.2%	70.7%
19	70.1%	72.1%	**71.2%**
20	70.3%	71.8%	69.1%
21	68.6%	71.7%	70.1%
22	67.3%	72.6%	69.8%

Table E.11: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22.

MOCHA PLP LDM results			
dimension	classification accuracy		
	PLP	PLP $+ \delta$	PLP $+ \delta + \delta\delta$
0	70.6%	73.4%	71.5%
1	70.7%	73.2%	71.6%
2	70.6%	72.9%	72.0%
3	70.5%	72.9%	70.3%
4	70.2%	73.1%	71.1%
5	71.0%	73.4%	71.8%
6	71.2%	73.0%	71.7%
7	71.7%	73.1%	71.1%
8	70.6%	72.9%	71.6%
9	**72.0%**	72.6%	70.5%
10	71.8%	72.9%	71.3%†
11	71.5%	72.5%†	70.4%
12	71.5%	71.5%	71.8%
13	71.4%†	**73.9%**	71.0%
14	71.2%	72.2%	71.8%
15	71.2%	73.2%	**72.2%**
16	70.6%	72.3%	71.8%
17	71.5%	72.2%	71.8%
18	71.0%	73.0%	70.5%
19	70.6%	72.8%	71.0%
20	70.8%	72.5%	71.8%

Table E.12: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20. These results are shown graphically in figure 5.8 on page 133

MOCHA MFCC factor analyser results			
dimension	classification accuracy		
	MFCC	MFCC $+ \delta$	MFCC $+ \delta + \delta\delta$
1	66.2%	68.1%	69.5%
2	68.4%	69.4%	71.6%
3	68.1%	71.6%	72.5%
4	69.2%	72.4%	71.8%
5	70.0%	73.6%	72.4%
6	68.8%	73.7%	72.9%
7	70.0%	73.6%	73.2%
8	**70.7%**	73.0%	74.3%
9	70.3%	73.8%	74.3%
10	70.6%	74.0%	74.2%
11	70.4%	74.3%	74.4%
12	70.4%	73.9%	74.2%
13	70.6%	73.7%	74.3%
14	70.5%	**74.5%**	75.3%
15	70.4%	74.2%	74.8%
16	70.6%	74.0%	75.0%
17	70.6%	73.7%	75.0%
18	70.6%	74.4%	75.5%
19	70.5%	74.5%	75.1%
20	70.6%	74.2%	**76.2%**
21	68.4%	74.0%	75.4%
22	68.4%	74.5%	75.5%

Table E.13: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22.

MOCHA MFCC diagonal covariance LDM results			
dimension	classification accuracy		
	MFCC	MFCC + δ	MFCC + $\delta + \delta\delta$
0	62.7%	64.9%	61.5%
1	66.0%	65.6%	62.5%
2	66.2%	68.4%	65.5%
3	67.8%	68.9%	67.9%
4	67.1%	69.1%	68.1%
5	66.4%	68.9%	69.0%
6	67.4%	69.3%	68.8%
7	67.5%	69.6%	69.0%
8	66.0%	68.7%	69.0%
9	68.2%	69.2%	69.0%
10	67.6%	69.0%	69.2%
11	67.6%	69.4%	68.8%
12	**68.9%**	68.1%	**69.7%**
13	68.4%	69.7%	68.2%
14	68.1%	**70.6%**	69.5%
15	67.3%	69.9%	68.5%
16	67.9%	69.0%	68.7%
17	67.1%	70.4%	69.2%
18	68.9%	69.3%	69.0%
19	68.0%	68.9%	66.7%
20	67.3%	70.2%	68.9%

Table E.14: Speaker-dependent classification accuracies for systems with diagonal covariance LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

MOCHA MFCC diagonal state covariance LDM results			
dimension	classification accuracy		
	MFCC	MFCC $+\delta$	MFCC $+\delta+\delta\delta$
1	70.5%	74.7%	75.4%
2	70.3%	75.2%	76.0%
3	70.7%	74.8%	75.1%
4	**71.9%**	75.1%	76.2%
5	68.3%	75.7%	75.3%
6	71.9%	75.4%	75.6%
7	70.7%	76.2%	75.3%
8	70.7%	**76.5%**	75.4%
9	70.7%	76.1%	75.4%
10	70.1%	75.3%	74.8%
11	70.0%	74.0%	**76.5%**
12	69.3%	76.4%	76.2%
13	71.2%	74.8%	76.2%
14	66.7%	76.1%	75.1%
15	71.2%	75.4%	75.9%
16	70.4%	74.5%	74.8%
17	71.5%	74.0%	75.9%
18	71.2%	75.5%	75.7%
19	70.9%	74.5%	74.6%
20	70.5%	75.2%	76.1%

Table E.15: Speaker-dependent classification accuracies for systems with diagonal state covariance LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

MOCHA MFCC state covariance identity LDM results			
dimension	classification accuracy		
	MFCC	MFCC $+\delta$	MFCC $+\delta+\delta\delta$
1	67.0%	75.1%	75.7%
2	69.9%	74.3%	75.4%
3	**70.2%**	**75.3%**	74.7%
4	70.4%	74.2%	75.6%
5	68.9%	74.0%	75.1%
6	70.1%	75.0%	75.2%
7	67.9%	73.2%	75.6%
8	67.9%	74.3%	75.3%
9	68.4%	74.2%	75.7%
10	67.9%	74.1%	**75.9%**
11	65.3%	73.2%	73.5%
12	67.8%	70.6%	75.1%
13	67.2%	73.6%	74.2%
14	64.9%	72.9%	75.1%
15	65.7%	73.7%	74.8%
16	67.0%	73.6%	74.6%
17	65.1%	73.2%	74.7%
18	63.4%	71.8%	73.7%
19	64.2%	72.3%	75.1%
20	66.8%	71.6%	73.6%

Table E.16: Speaker-dependent classification accuracies for systems with identity state covariance LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

MOCHA MFCC LDM results			
dimension	classification accuracy		
	MFCC	MFCC $+\,\delta$	MFCC $+\,\delta+\delta\delta$
0	70.7%	74.6%	75.4%
1	70.4%	74.8%	**76.2%**
2	71.1%	75.3%	75.6%
3	71.1%	74.4%	75.4%
4	71.2%	74.7%	75.9%
5	71.1%	76.2%	76.0%
6	71.4%	75.6%	74.8%
7	71.4%	75.3%†	75.3%
8	71.4%	76.0%	75.7%
9	71.5%	**76.4%**	75.6%†
10	**71.9%**	74.0%	75.1%
11	71.8%	75.3%	75.1%
12	71.8%	75.2%	74.3%
13	71.5%	75.9%	75.3%
14	71.7%	75.1%	75.8%
15	71.3%	76.4%	73.1%
16	71.2%	75.2%	75.0%
17	70.9%†	75.1%	75.6%
18	71.7%	74.2%	75.7%
19	70.5%	74.7%	75.2%
20	70.3%	74.6%	74.8%

Table E.17: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20. These results are shown graphically in figure 5.9 on page 134.

MOCHA combined PLP and EMA factor analyser results				
state dimension	classification accuracy			
	raw		LDA	
	PLP + EMA	PLP + EMA + δ	PLP + EMA + δ	PLP + EMA+ δ + $\delta\delta$s
1	70.7%	73.3%	75.4%	77.2%
2	68.4%	72.8%	75.6%	77.3%
3	72.5%	74.9%	76.6%	78.4%
4	73.0%	77.3%	77.5%	78.2%
5	74.3%	76.8%	77.5%	78.8%
6	73.0%	78.2%	77.6%	78.7%
7	74.2%	77.6%	77.5%	78.9%
8	**75.3%**	78.5%	79.0%	79.5%
9	73.0%	78.5%	79.0%	79.4%
10	73.1%	78.8%	78.5%	79.7%
11	73.2%	79.2%	78.1%	80.1%
12	73.1%	78.2%	**80.1%**	79.5%
13	72.5%	78.5%	79.4%	79.8%
14	72.6%	79.2%	79.0%	80.1%
15	73.3%	78.2%	79.3%	79.8%
16	73.3%	78.2%	79.6%	79.8%
17	73.7%	78.5%	79.5%	80.2%
18	72.3%	79.4%	79.0%	79.8%
19	71.2%	79.3%	79.6%	80.2%
20	71.9%	79.8%	78.9%	79.9%
21	71.7%	78.8%	79.8%	79.9%
22	73.5%	80.1%	79.0%	80.0%
23	72.5%	**80.4%**	80.0%	80.3%
24	72.5%	80.0%	79.3%	80.5%
25	71.1%	79.6%	79.3%	**80.7%**
26	73.0%	79.3%	79.4%	80.3%
27	71.2%	80.0%	79.7%	80.6%
28	72.5%	79.8%	79.5%	80.4%

Table E.18: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were combinations of PLPs and real EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 28.

MOCHA combined PLP and EMA LDM results				
state dimension	classification accuracy			
	raw		LDA	
	PLP + EMA	PLP + EMA + δ	PLP + EMA + δ	PLP + EMA+ δ + $\delta\delta$s
0	74.3%	79.8%	79.3%	79.1%
1	74.6%	79.5%	79.5%	80.3%
2	75.1%	79.7%	78.4%	80.0%
3	75.3%	79.0%	79.8%	79.0%
4	75.4%	79.6%	79.5%	80.4%
5	74.5%	80.3%	80.3%	80.5%
6	75.0%	80.5%	79.5%	80.6%
7	75.5%	80.3%	79.3%	80.6%
8	76.7%	81.0%	80.6%	**81.7%**
9	77.0%	80.3%	80.4%	80.9%
10	75.3%	81.0%	80.3%	80.8%
11	74.4%	79.4%	**81.1%**	80.2%
12	75.4%	80.2%	80.1%	80.7%
13	75.7%	80.7%	78.5%	80.7%
14	76.4%	80.7%	79.8%	80.2%
15	77.9%	79.8%	79.6%	80.2%
16	77.9%	80.5%	79.9%	80.4%
17	**79.3%**	80.7%	79.0%	81.0%
18	78.3%	80.6%	80.0%	80.7%
19	77.6%	80.6%	79.6%	80.8%
20	77.7%	80.9%	80.0%	80.9%
21	77.9%	80.2%	80.5%	79.9%
22	77.5%	79.2%	79.8%	79.8%
23	78.2%	80.4%	79.6%	81.2%
24	77.8%	**81.5%**	79.6%	81.1%
25	77.5%	80.6%	80.1%	80.6%
26	78.1%	79.3%	80.0%	80.7%
27	77.9%	80.0%	80.1%	80.7%
28	77.8%	80.7%	80.3%	80.9%

Table E.19: Speaker-dependent classification accuracies for systems with LDMs as the acoustic model. The features were combinations of PLPs and real EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 28. These results are shown pictorially in figure 5.12 on page 138

MOCHA combined MFCC and EMA factor analyser results				
state dimension	classification accuracy			
	raw		LDA	
	MFCC + EMA	MFCC + EMA + δ	MFCC + EMA + δ	MFCC + EMA+ δ + $\delta\delta$s
1	70.9%	75.3%	76.0%	75.4%
2	70.9%	75.6%	76.7%	76.2%
3	71.8%	75.0%	75.2%	77.3%
4	73.6%	76.0%	76.7%	77.9%
5	75.5%	76.1%	77.5%	79.1%
6	75.9%	77.9%	77.8%	79.3%
7	75.4%	78.5%	78.5%	79.6%
8	75.7%	78.5%	78.2%	79.6%
9	75.2%	76.8%	**79.2%**	78.9%
10	75.9%	78.1%	79.1%	78.9%
11	75.6%	78.6%	77.6%	79.3%
12	75.5%	78.5%	78.4%	**79.7%**
13	**76.1%**	78.3%	78.7%	78.6%
14	75.7%	79.0%	79.2%	78.9%
15	75.6%	79.0%	78.7%	79.1%
16	75.9%	78.5%	78.4%	79.3%
17	75.9%	78.9%	78.4%	79.0%
18	75.6%	79.1%	79.2%	78.9%
19	75.9%	79.2%	79.1%	79.0%
20	75.8%	**79.8%**	78.7%	78.9%
21	75.6%	79.2%	78.3%	76.3%
22	75.9%	79.5%	78.7%	76.2%

Table E.20: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were combinations of MFCCs and real EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 22.

MOCHA combined MFCC and EMA LDM results				
state dimension	classification accuracy			
	raw		LDA	
	MFCC + EMA	MFCC + EMA + δ	MFCC + EMA + δ	MFCC + EMA+ δ + $\delta\delta$s
0	75.3%	79.8%	79.5%	79.4%
1	75.0%	79.6%	78.7%	79.5%
2	75.3%	79.3%	79.3%	79.3%
3	74.5%	79.5%	78.7%	79.3%
4	77.1%	80.1%	79.8%	79.5%
5	76.7%	78.7%	79.6%	79.5%
6	76.8%	80.2%	79.3%	79.0%
7	76.4%	80.1%	79.4%	79.9%
8	77.2%	79.5%	79.0%	79.5%
9	77.8%	80.2%	79.2%	79.3%
10	77.0%	78.8%	79.1%	79.3%
11	77.0%	79.7%	79.9%	79.5%
12	78.1%	79.3%	80.3%	79.5%
13	77.2%	79.4%	80.4%	79.3%
14	78.0%	80.3%	79.2%	**80.3%**
15	78.1%	79.2%	79.8%	79.9%
16	77.6%	78.5%	**80.6%**	79.9%
17	**78.7%**	79.5%	79.2%	79.8%
18	77.8%	79.5%	79.9%	79.8%
19	78.1%	78.9%	78.8%	79.8%
20	78.3%	78.7%	80.1%	79.8%
21	76.9%	79.3%	80.1%	80.1%
22	77.6%	**80.4%**	79.8%	79.8%
23	77.0%	78.9%	80.0%	80.0%
24	75.4%	78.9%	78.7%	79.7%

Table E.21: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were combinations of MFCCs and real EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 24. These results are shown pictorially in figure 5.13 on page 139.

MOCHA combined PLP and network-recovered EMA factor analyser results				
state dimension	classification accuracy			
	raw		LDA	
	PLP + NET	PLP + NET + δ	PLP + NET + δ	PLP + NET+ δ + $\delta\delta$s
1	60.9%	56.6%	68.0%	67.9%
2	60.9%	61.4%	68.9%	70.0%
3	63.4%	62.5%	68.9%	69.8%
4	63.6%	63.7%	69.6%	69.7%
5	64.9%	64.1%	69.3%	70.1%
6	67.0%	64.4%	68.9%	69.4%
7	68.4%	64.9%	69.7%	71.2%
8	69.2%	65.4%	70.4%	71.4%
9	69.2%	67.6%	70.9%	71.1%
10	68.7%	68.0%	70.3%	71.3%
11	68.9%	65.6%	70.5%	70.9%
12	68.5%	66.8%	71.3%	69.8%
13	69.3%	66.4%	71.4%	69.9%
14	**69.7%**	66.7%	70.8%†	70.3%
15	69.0%	67.2%	71.1%	71.0%
16	69.4%	66.7%	71.0%	71.2%
17	69.0%	68.0%	70.6%	70.2%
18	69.3%	68.5%	71.2%	71.1%†
19	69.2%	67.6%	71.3%	70.9%
20	69.5%†	68.6%	70.8%	**72.2%**
21	68.6%	68.1%	70.9%	70.3%
22	68.6%	68.2%†	71.2%	71.4%
23	69.3%	67.7%	71.6%	70.7%
24	69.0%	68.8%	71.9%	71.1%
25	69.2%	68.1%	72.0%	71.2%
26	68.6%	**69.5%**	**72.5%**	70.6%
27	69.4%	68.7%	71.8%	70.4%
28	69.5%	69.1%	71.5%	71.4%

Table E.22: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were combinations of PLPs and recovered EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 28.

MOCHA combined PLP and network-recovered EMA LDM results				
state dimension	classification accuracy			
	raw		LDA	
	PLP + NET	PLP + NET + δ	PLP + NET + δ	PLP + NET+ δ + $\delta\delta$s
0	70.2%	71.6%	71.5%	72.0%
1	70.8%	72.8%	71.6%	73.2%
2	70.3%	72.8%	72.5%	72.2%
3	70.9%	73.7%	72.0%	72.0%
4	70.0%	73.8%	72.8%	73.1%
5	71.2%	73.1%	71.1%	72.6%
6	70.3%	72.9%	72.3%	73.0%
7	71.2%	72.6%	72.8%	72.1%
8	70.8%	72.9%	72.7%	72.0%
9	71.7%	72.6%	72.2%	**73.6%**
10	72.3%	72.3%	72.8%	72.8%
11	70.7%	72.5%	72.0%	71.4%
12	72.2%	**74.3%**†	72.0%	72.3%
13	**72.5%**	72.7%	72.2%	72.0%
14	72.3%	72.4%	**73.4%**†	72.3%
15	72.0%	72.3%	71.5%	71.8%
16	72.2%	72.9%	72.0%	72.0%
17	71.7%	72.5%	71.8%	70.9%
18	71.7%	72.7%	72.3%	71.9%†
19	71.6%	72.3%	72.5%	72.2%
20	71.7%†	72.5%	71.7%	71.5%
21	72.3%	72.0%	73.4%	71.8%
22	72.1%	73.1%	72.4%	72.2%
23	71.1%	72.9%	71.7%	72.2%
24	70.0%	72.9%	71.2%	71.7%

Table E.23: Speaker-dependent classification accuracies for systems with LDMs as the acoustic model. The features were combinations of PLPs and network-recovered EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 24. These results are shown graphically in figure 5.16 on page 144

MOCHA combined MFCC and network-recovered EMA factor analyser results				
state dimension	classification accuracy			
	raw		LDA	
	MFCC + NET	MFCC + NET + δ	MFCC + NET + δ	MFCC + NET+ δ + $\delta\delta$s
1	59.9%	58.4%	68.3%	68.3%
2	61.5%	63.7%	69.4%	69.8%
3	62.2%	64.9%	71.0%	70.5%
4	62.5%	64.0%	70.2%	69.5%
5	64.7%	67.0%	70.5%	69.3%
6	66.4%	66.3%	69.8%	69.1%
7	68.1%	67.2%	71.8%	69.9%
8	68.3%	66.7%	71.7%	71.0%
9	68.2%	67.8%	71.2%	70.6%
10	67.8%	68.3%	71.2%	71.5%
11	68.9%	68.6%	71.5%	70.3%
12	67.9%	69.7%	71.4%	70.1%
13	68.6%	70.1%	72.0%	70.1%
14	68.5%	69.2%	72.2%	69.5%
15	67.9%	69.8%	71.9%	69.7%
16	68.8%	69.8%	72.0%	71.1%
17	**69.5%**	70.6%	72.5%	70.3%
18	68.5%	71.4%	72.0%	70.1%
19	69.1%	70.6%	**72.8%**†	71.5%
20	68.7%†	70.3%	72.6%	70.6%
21	67.1%	71.0%	72.5%	71.0%
22	67.8%	70.8%	72.6%	70.1%
23	67.8%	71.1%	72.5%	71.5%
24	67.7%	71.1%	71.7%	70.7%†
25	67.3%	71.1%	72.6%	71.8%
26	68.1%	70.9%	72.8%	71.5%
27	67.5%	70.9%	72.3%	**72.5%**
28	67.9%	71.4%	72.5%	72.3%
29	67.7%	**71.5%**†	72.8%	72.5%
30	67.7%	71.1%	72.8%	71.2%
31	68.1%	71.2%	72.5%	71.5%
32	68.3%	71.2%	72.2%	72.0%

Table E.24: Speaker-dependent classification accuracies for systems with factor analysers used as the acoustic model. The features were combinations of MFCCs and network-recovered EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 32.

MOCHA combined MFCC and network-recovered EMA LDM results				
state dimension	classification accuracy			
	raw		LDA	
	MFCC + NET	MFCC + NET + δ	MFCC + NET + δ	MFCC + NET+ δ + $\delta\delta$s
0	69.1%	72.4%	71.8%	72.0%
1	70.7%	72.2%	71.5%	70.9%
2	69.8%	73.6%	72.2%	72.0%
3	70.1%	72.1%	72.6%	71.9%
4	70.4%	72.8%	72.4%	71.8%
5	70.0%	73.8%	73.4%	72.5%
6	71.1%	73.3%	73.3%	71.5%
7	70.9%	74.1%	72.4%	72.3%
8	71.6%	74.1%	72.9%	73.1%
9	69.9%	73.6%	73.6%	72.2%
10	71.2%	74.0%	73.0%	70.9%
11	70.9%	73.7%	73.1%	72.8%
12	71.0%	73.8%	72.7%	71.8%
13	71.1%	73.7%	72.2%	72.7%
14	71.6%	74.0%	73.6%	71.2%
15	70.7%	75.0%	72.0%	71.3%
16	**72.8%**	**75.2%**	73.1%†	72.5%
17	71.0%	73.9%	73.0%	72.4%
18	71.5%	74.3%	73.6%	73.0%
19	71.7%	74.7%†	73.1%	72.2%
20	71.1%	74.7%	74.7%	72.0%
21	72.0%†	73.5%	72.9%	72.2%
22	71.7%	74.5%	73.4%	72.3%
23	71.1%	74.0%	73.9%	71.5%
24	72.3%	73.7%	72.8%	73.0%
25	72.2%	73.4%	**74.9%**	72.7%
26	71.5%	74.6%	72.8%	72.8%
27	72.1%	73.8%	73.5%	73.2%†
28	71.8%	73.9%	73.0%	72.3%
29	72.5%	74.9%	74.6%	71.3%
30	72.4%	75.1%	72.6%	**73.4%**

Table E.25: Speaker-dependent classification accuracies for systems with LDMs used as the acoustic model. The features were combinations of MFCCs and recovered EMA data, used raw or post-processed using LDA. Accuracies are given for a state dimensions ranging from 0 to 30. These results are shown graphically in figure 5.17 on page 145

TIMIT PLP factor analyser 61 phone results			
state dimension	61 phone classification accuracy		
	PLP	PLP + δ	PLP + δ + $\delta\delta$
1	52.5%	56.6%	58.2%
2	54.2%	57.3%	58.9%
3	54.4%	58.3%	60.1%
4	55.0%	59.1%	60.4%
5	55.5%	59.3%	61.0%
6	55.9%	59.8%	61.3%
7	56.0%	59.7%	61.7%
8	56.6%	59.8%	62.0%
9	57.0%	60.4%	62.1%
10	57.2%	60.5%	62.4%
11	57.2%	60.5%	62.6%
12	57.2%	60.6%	62.6%
13	57.2%†	60.8%	62.6%
14	57.2%	60.5%	62.6%
15	**57.3%**	60.8%	62.7%
16	57.2%	60.9%	62.8%
17	57.2%	60.9%	62.8%
18	57.2%	**61.0%**	**62.9%**
19	57.2%	61.0%	62.8%
20	57.2%	61.0%†	62.8%†
21	56.2%	61.0%	62.8%
22	56.5%	60.9%	62.8%

Table E.26: Speaker-independent classification accuracies for systems with factor analysers used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22

TIMIT PLP factor analyser 39 phone results			
state dimension	39 phone classification accuracy		
	PLP	PLP + δ	PLP + δ + $\delta\delta$
1	62.2%	65.3%	66.9%
2	63.8%	65.8%	67.6%
3	63.9%	66.9%	68.5%
4	64.3%	67.8%	68.7%
5	64.8%	67.7%	69.2%
6	65.1%	68.0%	69.6%
7	65.3%	68.2%	69.9%
8	65.9%	68.1%	70.2%
9	66.1%	68.6%	70.2%
10	66.3%	68.7%	70.4%
11	66.4%	68.9%	70.7%
12	66.4%	68.8%	70.7%
13	**66.5%**†	69.1%	70.7%
14	66.4%	68.7%	70.7%
15	66.5%	69.1%	70.8%
16	66.4%	69.2%	70.8%
17	66.5%	**69.3%**	**70.9%**
18	66.4%	69.3%	70.9%
19	66.5%	69.3%	70.9%
20	66.4%	69.3%†	70.9%†
21	65.4%	69.3%	70.8%
22	65.7%	69.2%	70.9%

Table E.27: Speaker-independent classification accuracies for systems with factor analysers used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22

TIMIT PLP LDM 61 phone results			
state dimension	61 phone classification accuracy		
	PLP	PLP + δ	PLP + δ + $\delta\delta$
0	57.2%	62.1%	63.1%
1	57.1%	62.1%	63.2%
2	57.7%	61.7%	63.1%
3	58.1%	62.3%	63.6%
4	58.7%	61.8%	63.8%
5	58.4%	61.6%	63.3%
6	58.7%	62.1%	63.6%
7	58.9%	61.9%	63.3%
8	59.2%	61.9%	63.5%
9	58.9%	62.4%†	63.4%
10	**59.3%**†	61.5%	63.5%
11	59.2%	62.2%	63.2%
12	58.9%	62.4%	63.6%
13	59.3%	61.9%	**64.0%**†
14	59.1%	62.1%	63.6%
15	59.0%	62.1%	63.9%
16	59.0%	62.2%	63.7%
17	57.0%	61.8%	63.7%
18	57.2%	61.8%	63.7%
19	57.5%	61.8%	63.3%
20	57.8%	**62.6%**	63.9%
21	57.8%	62.1%	63.4%
22	58.1%	62.1%	63.8%

Table E.28: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22

TIMIT PLP LDM 39 phone results			
state dimension	39 phone classification accuracy		
	PLP	PLP + δ	PLP + δ + $\delta\delta$
0	66.3%	70.1%	71.3%
1	66.2%	70.2%	71.5%
2	66.7%	70.1%	71.4%
3	67.0%	70.5%	71.9%
4	67.4%	70.2%	71.9%
5	67.2%	70.1%	71.5%
6	67.4%	70.4%	71.7%
7	67.7%	70.4%	71.5%
8	**67.8%**	70.6%	71.7%
9	67.6%	**71.0%**†	71.8%
10	67.8%†	70.3%	71.8%
11	67.8%	70.5%	71.5%
12	67.6%	70.7%	71.7%
13	67.8%	70.5%	**72.2%**†
14	67.8%	70.7%	71.8%
15	67.8%	70.8%	72.1%
16	67.7%	70.6%	72.0%
17	66.6%	70.5%	71.9%
18	66.6%	70.5%	72.0%
19	66.7%	70.6%	71.6%
20	67.1%	70.8%	72.1%
21	67.0%	70.7%	71.7%
22	67.1%	70.8%	72.1%

Table E.29: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features were PLPs, PLPs with δ coefficients, and PLPs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 22. These results are shown graphically in figure 5.18 on page 151

TIMIT MFCC factor analyser 61 phone results			
state dimension	61 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	52.6%	56.6%	58.3%
2	53.7%	57.5%	58.9%
3	54.6%	58.2%	60.1%
4	55.0%	58.8%	60.4%
5	55.5%	59.0%	60.5%
6	56.0%	59.4%	61.3%
7	56.6%	59.7%	61.1%
8	56.7%	60.2%	61.7%
9	57.1%	60.5%	61.6%
10	**57.2%**	60.5%	61.9%
11	57.2%†	60.7%	62.3%
12	57.1%	60.9%	62.2%
13	57.1%	60.9%	62.5%
14	57.1%	61.2%	62.6%
15	57.1%	61.5%	62.6%
16	57.1%	61.5%	62.6%
17	57.1%	61.5%	62.6%
18	57.1%	61.5%	62.6%
19	57.2%	61.7%	62.8%†
20	57.1%	**61.8%**†	62.8%
21	56.7%	61.0%	62.9%
22	56.7%	61.1%	62.8%
23	56.7%	61.2%	63.0%
24	56.7%	61.2%	62.6%
25	56.7%	61.2%	62.9%
26	56.8%	61.2%	**63.2%**

Table E.30: Speaker-independent classification accuracies for systems with factor analysers used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 26.

TIMIT MFCC factor analyser 39 phone results			
state dimension	39 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	62.1%	65.6%	67.0%
2	63.3%	66.3%	67.4%
3	64.0%	66.9%	68.5%
4	64.4%	67.3%	68.6%
5	64.8%	67.5%	68.8%
6	65.2%	67.8%	69.5%
7	65.7%	68.1%	69.4%
8	65.8%	68.6%	69.9%
9	66.2%	68.7%	69.8%
10	**66.3%**	68.8%	70.0%
11	66.3%†	68.9%	70.3%
12	66.2%	69.1%	70.2%
13	66.2%	69.2%	70.4%
14	66.2%	69.4%	70.7%
15	66.2%	69.7%	70.6%
16	66.2%	69.7%	70.7%
17	66.3%	69.7%	70.7%
18	66.2%	69.8%	70.6%
19	66.3%	69.9%	70.7%†
20	66.2%	**70.0%**†	70.8%
21	65.8%	69.3%	70.9%
22	65.8%	69.4%	70.8%
23	65.8%	69.4%	71.0%
24	65.9%	69.5%	71.0%
25	65.8%	69.5%	70.9%
26	65.9%	69.5%	**71.3%**

Table E.31: Speaker-independent classification accuracies for systems with factor analysers used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 26.

TIMIT MFCC diagonal covariance LDM 61 phone results			
state dimension	61 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	52.2%	56.0%	58.1%
2	53.3%	57.5%	57.3%
3	53.1%	58.4%	59.0%
4	53.0%	59.6%	59.5%
5	53.4%	59.0%	59.5%
6	54.4%	59.4%	59.6%
7	53.9%	58.8%	59.9%
8	54.2%	59.4%	59.9%
9	55.4%†	59.1%	60.0%
10	54.4%	59.5%	**60.7%**†
11	54.0%	59.6%†	60.3%
12	54.7%	59.4%	60.4%
13	**55.7%**	59.3%	60.4%
14	54.5%	58.9%	59.7%
15	55.6%	59.4%	60.6%
16	54.6%	59.6%	60.7%
17	55.4%	59.7%	60.3%
18	55.3%	59.7%	60.2%
19	55.3%	**59.8%**	60.0%
20	55.5%	59.6%	59.8%

Table E.32: Speaker-independent classification accuracies for systems with diagonal covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

TIMIT MFCC diagonal covariance LDM 39 phone results			
state dimension	39 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	62.0%	65.0%	66.7%
2	62.7%	66.6%	66.3%
3	62.5%	67.5%	67.9%
4	62.3%	68.3%	68.3%
5	62.9%	67.9%	68.5%
6	63.7%	68.2%	68.4%
7	63.3%	67.6%	68.7%
8	63.7%	68.5%	68.7%
9	64.6%†	68.1%	69.0%
10	63.9%	68.5%	69.6%†
11	63.6%	68.7%†	69.1%
12	64.3%	68.3%	69.2%
13	**65.1%**	68.3%	69.5%
14	64.1%	68.2%	69.0%
15	64.9%	68.5%	**69.9%**
16	64.2%	68.6%	69.4%
17	64.9%	68.7%	69.5%
18	64.8%	**68.8%**	69.4%
19	64.8%	68.8%	69.4%
20	64.9%	68.7%	69.3%

Table E.33: Speaker-independent classification accuracies for systems with diagonal covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

TIMIT MFCC state diagonal covariance LDM 61 phone results			
state dimension	61 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	57.5%	62.0%	63.3%
2	57.4%	62.4%	63.5%
3	57.4%	61.8%	63.7%
4	57.6%	62.1%	**64.0%**
5	57.2%	62.1%	63.8%
6	57.9%	62.1%	63.9%
7	**58.4%**	61.8%	63.8%
8	58.1%†	61.6%	63.5%
9	58.3%	62.7%	63.6%
10	58.0%	62.6%	63.7%
11	57.5%	62.2%	63.2%
12	58.1%	62.6%	63.7%
13	56.9%	62.5%	63.6%
14	57.9%	62.5%	63.7%
15	58.1%	62.6%	63.8%
16	57.5%	62.7%	63.8%
17	58.2%	**62.8%**†	63.9%†
18	58.1%	62.8%	63.9%
19	58.3%	62.4%	63.8%
20	58.1%	61.9%	63.7%

Table E.34: Speaker-independent classification accuracies for systems with state diagonal covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

TIMIT MFCC state diagonal covariance LDM 39 phone results			
state dimension	39 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	66.6%	70.3%	71.4%
2	66.6%	70.7%	71.7%
3	66.6%	70.4%	71.7%
4	66.7%	70.5%	72.1%
5	66.6%	70.6%	72.0%
6	67.1%	70.5%	72.0%
7	67.4%	70.3%	72.0%
8	67.2%†	70.3%	71.7%
9	67.3%	71.1%	71.9%
10	67.2%	71.1%	72.0%
11	66.9%	71.0%	71.5%
12	67.3%	71.1%	72.0%
13	66.2%	70.9%	71.9%
14	67.0%	71.1%	72.0%
15	67.3%	71.1%	72.2%
16	66.8%	71.2%	72.2%
17	67.3%	**71.5%**†	72.2%†
18	67.4%	71.2%	**72.3%**
19	**67.5%**	71.0%	72.2%
20	67.2%	70.6%	72.2%

Table E.35: Speaker-independent classification accuracies for systems with state diagonal covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

TIMIT MFCC state identity covariance LDM 61 phone results			
state dimension	61 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	57.3%	62.0%	63.5%
2	57.3%	61.8%	63.4%
3	57.6%	62.1%	63.6%
4	57.6%	62.2%	63.4%
5	57.4%	61.9%	63.6%
6	57.8%	62.4%	63.3%
7	57.9%	61.9%	63.6%
8	57.6%	62.1%	63.5%
9	**58.1%**	62.4%	63.4%
10	57.5%	61.9%	63.5%
11	58.1%	62.2%	**63.7%**
12	58.0%	61.9%	63.5%
13	56.4%	62.3%	63.5%
14	57.7%	62.6%	63.6%
15	58.0%	**62.7%**	63.3%
16	58.1%	62.1%	63.2%
17	57.9%	62.4%	63.5%
18	58.1%	62.3%	63.7%
19	57.9%†	62.3%	63.5%†
20	57.8%	62.4%†	63.4%
21	57.9%	62.6%	63.4%
22	57.8%	62.1%	63.7%
23	57.9%	62.6%	63.2%
24	57.7%	**62.8%**	63.5%

Table E.36: Speaker-independent classification accuracies for systems with state identity covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 24.

TIMIT MFCC state identity covariance LDM 39 phone results			
state dimension	39 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
1	66.5%	70.3%	71.6%
2	66.5%	70.2%	71.6%
3	66.7%	70.6%	71.8%
4	66.7%	70.6%	71.7%
5	66.6%	70.5%	71.7%
6	66.9%	70.8%	71.6%
7	66.9%	70.5%	71.9%
8	66.7%	70.7%	71.9%
9	67.2%	70.9%	71.8%
10	66.7%	70.6%	71.8%
11	**67.3%**	70.7%	72.0%
12	67.1%	70.6%	71.8%
13	65.6%	70.8%	71.8%
14	67.0%	**71.3%**	72.0%
15	67.1%	71.1%	71.7%
16	67.2%	70.7%	71.7%
17	67.2%	70.9%	72.0%
18	67.2%	70.8%	72.1%
19	67.0%†	70.9%	71.9%†
20	67.0%	71.0%†	71.8%
21	67.0%	71.2%	71.9%
22	67.1%	70.8%	**72.2%**
23	67.0%	71.1%	71.8%
24	67.0%	71.2%	71.9%

Table E.37: Speaker-independent classification accuracies for systems with state identity covariance LDMs as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 24.

TIMIT MFCC LDM 61 phone results			
state dimension	61 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
0	57.2%	62.0%	63.3%
1	57.9%	62.1%	63.7%
2	57.7%	62.2%	63.3%
3	57.8%	62.3%	63.4%
4	57.6%	62.4%	63.6%
5	57.8%	61.9%	63.8%
6	58.0%	62.2%	64.0%
7	57.5%	62.5%	63.6%
8	**58.5%**	62.7%	63.6%
9	57.7%	62.4%	**64.2%**†
10	57.5%	62.6%	63.8%
11	57.2%	62.5%	63.9%
12	58.4%†	**62.8%**†	64.1%
13	58.0%	62.6%	63.8%
14	57.9%	62.0%	63.8%
15	57.9%	62.0%	63.5%
16	58.1%	62.3%	63.7%
17	58.0%	62.2%	63.4%
18	58.2%	62.4%	63.4%
19	57.9%	62.3%	63.6%
20	58.1%	61.8%	63.4%

Table E.38: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20. These results are shown graphically in figure 5.19 on page 152

TIMIT MFCC LDM 39 phone results			
state dimension	39 phone classification accuracy		
	MFCC	MFCC + δ	MFCC + δ + $\delta\delta$
0	66.4%	70.2%	71.3%
1	66.7%	70.4%	71.8%
2	66.8%	70.5%	71.4%
3	66.9%	70.7%	71.6%
4	66.7%	71.0%	71.9%
5	66.9%	70.6%	71.9%
6	67.1%	70.9%	72.2%
7	66.8%	71.1%	71.8%
8	67.4%	71.2%	72.0%
9	66.9%	71.0%	72.3%†
10	66.8%	71.1%	72.2%
11	66.6%	71.0%	72.1%
12	**67.4%**†	**71.3%**†	**72.4%**
13	67.1%	71.2%	72.1%
14	67.0%	70.9%	72.1%
15	67.1%	70.9%	71.8%
16	67.2%	71.1%	71.9%
17	67.1%	71.0%	71.7%
18	67.2%	71.1%	71.7%
19	67.1%	71.1%	71.8%
20	67.3%	70.8%	71.8%

Table E.39: Speaker-independent classification accuracies for systems with LDMs used as the acoustic model. The features were MFCCs, MFCCs with δ coefficients, and MFCCs with δ and $\delta\delta$ coefficients. Accuracies are given for a state dimensions ranging from 0 to 20.

Bibliography

Balakrishnama, S. & Ganapathiraju, A. (1998), Linear discriminant analysis - a brief tutorial, Technical report, Institute for Signal and Information Processing, Mississippi State University.

Bengio, Y. & Granvalet, Y. (2003), No unbiased estimator of the variance of K-fold cross-validation., Technical Report 1234, Département d'informatique et recherche opérationnelle, Université de Montréal.

Blackburn, C. (1996), Articulatory Methods for Speech Production and Recognition, PhD thesis, University of Cambridge.

Blackburn, C. S. & Young, S. (1996), Pseudo-articulatory speech synthesis for recognition using automatic feature extraction from x-ray data, *in* 'Proc. ICSLP', Philadelphia.

Blackburn, C. S. & Young, S. (2000), 'A self-learning predictive model of articulator movements during speech production', *Journal of the Acoustical Society of America* **107**, 1659–1670.

Blackburn, C. & Young, S. (1995), Towards improving speech recognition using a speech production model, *in* 'Proc. Eurospeech', Vol. 2, pp. 1623–1626.

Bourlard, H., Hermansky, H. & Morgan, N. (1996), 'Towards increasing speech recognition error rates', *Speech Communication* **18**, 205–231.

Breiman, L. & Friedman, J. H. (1985), 'Estimating optimal transformations for multiple regression and correlation.', *Journal of the American Statistical Association* **80**(391), 580–598.

Browman, C. & Goldstein, L. (1992), 'Articulatory phonology: an overview', *Phonetica* **49**, 155–180.

Carreira-Perpiñán, M. (1997), A review of dimension reduction techniques, Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield.

Chang, S., Greenberg, S. & Wester, M. (2001), An elitist approach to articulatory-acoustic feature classification, *in* 'Proc. Eurospeech', Aalborg, Denmark, pp. 1725–1728.

Chen, R. & Jamieson, L. (1996), Experiments on the implementation of recurrent neural networks for speech phone recognition, *in* 'Proc. Thirtieth Annual Asilomar Conference on Signals, Systems and Computers', Pacific Grove, California, pp. 779–782.

Chomsky, N. & Halle, M. (1968), *The Sound Pattern of English*, Harper & Row, New York, NY.

Clarkson, P. & Rosenfeld, R. (1997), Statistical language modelling using the CMU-cambridge toolkit, *in* 'Proc. Eurospeech'.

Cole, R., Noel, M., Lander, T. & Durham, T. (1995), New telephone speech corpora at CSLU, *in* 'Proc. Fourth European Conference on Speech Communication and Technology', Vol. 1, pp. 821–824.

Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete data via the EM algorithm (with discussion).', *Journal of the Royal Statistical Society* **B**(39), 1–38.

Deng, L. & Sun, D. (1994*a*), Phonetic classification and recognition using HMM representation of overlapping articulatory features for all classes of English sounds, *in* 'Proc. ICASSP', Vol. I, pp. 45–48.

Deng, L. & Sun, D. X. (1994*b*), 'A statistical framework for automatic speech recognition using the atomic units constructed from overlapping articulatory features', *Journal of the Acoustical Society of America* **95**(5), 2702–2719.

Digalakis, V. (1992), Segment-based stochastic models of spectral dynamics for continuous speech recognition, PhD thesis, Boston University Graduate School.

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

Digalakis, V., Ostendorf, M. & Rohlicek, J. (1989), Improvements in the stochastic segment model for phoneme recognition, *in* 'Proc. of the DARPA Speech and Natural Language Workshop', Cape Cod, Massachusetts, pp. 332–338.

Digalakis, V., Rohlicek, J. & Ostendorf, M. (1993), 'ML estimation of a stochastic linear system with the em algorithm and its application to speech recognition', *IEEE Trans. Speech and Audio Processing* **1**(4), 431–442.

Duda, R. & Hart, P. (1973), *Pattern Recognition and Scene Analysis*, John Wiley, New York, chapter 4.11.

Eide, E., Rohlicek, J., Gish, H. & Mitter, S. (1993), A linguistic feature representation of the speech waveform, *in* 'Proc. ICASSP-93', pp. 483–486.

Erler, K. & Freeman, G. (1996), 'An HMM-based speech recogniser using overlapping articulatory features', *Journal of the Acoustical Society of America* **100**, 2500–13.

Fitt, S. & Isard, S. (1999), Synthesis of regional English using a keyword lexicon., *in* 'Proc. Eurospeech', Vol. 2, pp. 823–6.

Frankel, J. & King, S. (2001*a*), ASR - articulatory speech recognition, *in* 'Proc. Eurospeech', Aalborg, Denmark, pp. 599–602.

Frankel, J. & King, S. (2001*b*), Speech recognition in the articulatory domain: investigating an alternative to acoustic HMMs, *in* 'Proc. Workshop on Innovations in Speech Processing'.

Frankel, J., Richmond, K., King, S. & Taylor, P. (2000), An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces, *in* 'Proc. ICLSP'.

Freitag, D. & McCallum, A. (2000), Information extraction with HMM structures learned by stochastic optimization, *in* 'AAAI/IAAI', pp. 584–589.

Gales, M. J. F. & Young, S. J. (1993), The theory of segmental hidden Markov models, Technical report, Cambridge University Engineering Department.

Ghahramani, Z. & Hinton, G. (1996*a*), Parameter estimation for linear dynamical systems, Technical Report CRG-TR-96-2, Dept of Computer Science, University of Toronto.

Ghahramani, Z. & Hinton, G. (1996*b*), Switching state-space models, Technical Report CRG-TR-96-3, Dept. Computer Science, University of Toronto.

Gish, H. & Ng, K. (1993), A segmental speech model with application to word spotting., *in* 'Proc. ICASSP', pp. 447–450.

Godfrey, J., Holliman, E. & McDaniel, J. (1992), Telephone speech corpus for research and development, *in* 'Proc. ICASSP', San Francisco.

Gold, B. & Morgan, N. (1999), *Speech and Audio Signal Processing*, Wiley Press.

Goldenthal, W. (1994), Statistical trajectory models for phonetic recognition., PhD thesis, M.I.T.

Hastie, T. & Tibshirani, R. (1986), 'Generalized additive models', *Statistical Science* **1**, 297–318.

Haykin, S., ed. (2001), *Kalman Filtering and Neural Networks*, Wiley Publishing.

Hermansky, H., Morgan, N., Bayya, A. & Kohn, P. (1991), Rasta-PLP speech analysis, Technical Report TR-91-069, ICSI, Berkeley, California.

Holmes, W. (1996), Modelling variability between and within speech segments for automatic speech recognition, *in* 'Speech Hearing and Language: work in progress 1996', Vol. 9, Department of Phonetics and Linguistics, UCL.

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

Holmes, W. J. & Russell, M. J. (1995), Speech recognition using a linear dynamic segmental HMM, *in* 'Proc. Eurospeech', pp. 1611–1614.

Iso, K. (1993), Speech recognition using dynamical model of speech production, *in* 'Proc. ICASSP', Minneapolis, MN, pp. II:283–286.

Iyer, R., Gish, H., Siu, M., Zavaliagkos, G. & Matsoukas, S. (1998), Hidden Markov models for trajectory modelling, *in* 'Proc. ICSLP'.

Iyer, R., Kimball, O. & Gish, H. (1999), Modelling trajectories in the HMM framework, *in* 'Proc. Eurospeech'.

Junqua, J. (1993), 'The Lombard reflex and its role on human listeners and automatic speech recognisers', *Journal of the Acoustical Society of America* **93**, 510–524.

Kalman, R. (1960), 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering* **82**, 35–44.

Keating, P. (1998), 'Word-level phonetic variation in large speech corpora', *ZAS Papers in Linguistics* **11**, 35–50.

Kenny, P., Hollan, R., Gupta, V., Lennig, M., Mermelstein, P. & O'Shaughnessy, D. (1993), 'A*-admissible heuristics for rapid lexical access.', *IEEE Transactions on Speech and Audio Processing.* **1**(1), 49–58.

Kimball, O. (1994), Segment modelling alternatives for continuous speech recognition, PhD thesis, E.C.S. Department, Boston University.

King, S., Stephenson, T., Isard, S., Taylor, P. & Strachan, A. (1998), Speech recognition via phonetically featured syllables, *in* 'Proc. ICSLP', pp. 1031–1034.

King, S. & Taylor, P. (2000), 'Detection of phonological features in continuous speech using neural networks', *Computer Speech and Language* **14**, 333–353.

King, S., Taylor, P., Frankel, J. & Richmond, K. (2000), Speech recognition via phonetically-featured syllables., *in* 'PHONUS', Vol. 5, Institute of Phonetics, University of the Saarland, pp. 15–34.

Kirchhoff, K. (1998), Robust Speech Recognition Using Articulatory Information, PhD thesis, Berkeley, CA.

Kirchhoff, K., Fink, G. & Sagerer, G. (2002), 'Combining acoustic and articulatory feature information for robust speech recognition', *Speech Communication* pp. 303–319.

Kohler, K., Lex, G., Patzold, M., Scheffers, M., Simpson, A. & Thon, W. (1994), Handbuch zur datenaufnahmen und transliteration in TP14 von VERBMOBIL - 3.0., Technical Report 11, IPDS Kiel.

Lamel, L. & Gauvain, J. (1993*a*), High performance speaker-independent phone recognition using CDHMM., *in* 'Proc. Eurospeech'.

Lamel, L. & Gauvain, J. (1993*b*), Identification of non-linguistic speech features, *in* 'Proc. Eurospeech', Berlin, pp. 23–30.

Lamel, L., Kassel, R. & Seneff, S. (1986), Speech database development: design and analysis of the acoustic-phonetic corpus., *in* 'Proc. Speech Recognition Workshop', Palo Alto, CA., pp. 100–109.

Lee, J. J., Kim, J. & Kim, J. (2001), 'Data-driven design of HMM topology for on-line handwriting recognition', *International Journal of Pattern Recognition and Artificial Intelligence* **15**(1), 107–121.

Lee, K. & Hon, H. (1989), 'Speaker-independent phone recognition using hidden Markov models.', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **37**(11), 1641–1648.

Liberman, A. & Mattingly, I. (1985), 'The motor theory of speech perception revisited', *Cognition* **21**, 1–36.

Lindblom, B., Lubker, J. & Gay, T. (1979), 'Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation', *Journal of Phonetics* **7**, 147–161.

Ljung, L. (1999), *System Identification - Theory For the User*, 2nd edn, PTR Prentice Hall, Upper Saddle River, N.J.

Macho, D., Nadeu, C., Jancovic, P. Rozinaj, G. & Hernando, J. (1999), Comparison of time and frequency filtering and cepstral-time matrix approaches in ASR, *in* 'Proc. Eurospeech', Budapest, Hungary, pp. 77–80.

Merhav, N. & Ephraim, Y. (1991), Hidden Markov modelling using the most likely state sequence, *in* 'IEEE Int. Conf. Acoust., Speech, Signal Processing', IEEE, pp. 469–472.

Mohri, M. (1997), 'Finite-state transducers in language and speech processing', *Computational Linguistics* **23**(2), 269–311.

Morgan, N. & Bourlard, H. (1995), 'Neural networks for statistical recognition of continuous speech', *Proceedings of the IEEE* **83**(5), 741–770.

Murphy, K. P. (1998), Switching Kalman filters, Technical report, University of California Berkeley.

Neal, R. & Hinton, G. (1998), *Learning in Graphical Models*, Kluwer Academic Publishers, chapter A view of the EM algorithm that justifies incremental, sparse, and other variants, pp. 355–368.

Nilsson, N. J. (1971), *Problem-Solving Methods in Artificial Intelligence*, MacGraw-Hill (New York NY).

Ostendorf, M. (1999), Moving beyond the 'beads-on-a-string' model of speech, *in* 'Proc. IEEE ASRU Workshop'.

Ostendorf, M. & Digalakis, V. (1991), The stochastic segment model for continuous speech recognition, *in* 'Proc. of the 25th Asilomar Conference on Signals, Systems and Computers', pp. 964–968.

Ostendorf, M., Digalakis, V. & Kimball, O. (1996), 'From HMMs to segment models: A unified view of stochastic modelling for speech recognition.', *IEEE Trans. on Speech and Audio Processing* .

Ostendorf, M. & Singer, H. (1997), 'HMM topology design using maximum likelihood successive state splitting', *Computer Speech and Language* **11**(1), 17–41.

Papcun, G., Hochberg, J., Thomas, T. R., Laroche, F., Zachs, J. & Levy, S. (1992), 'Inferring articulation and recognising gestures from acoustics with a neural network trained on x-ray microbeam data', *Journal of the Acoustical Society of America* **92**(2), 688–700.

Paul, D. (1992.), An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model., *in* 'Proc. ICASSP', Vol. 1, San Francisco, pp. 25–28.

Picone, J., Pike, S., Regan, R., Kamm, T., Bridle, J., Deng, L., Ma, Z., Richards, H. & Schuster, M. (1999), Initial evaluation of hidden dynamic models on conversational speech, *in* 'Proc. ICASSP', Phoenix, Arizona.

Price, P., Fisher, W. M., Bernstein, J. & Pallett, D. S. (1988), The DARPA 1000-word resource management database for continuous speech recognition., *in* 'Proc. ICASSP', Institute of Electrical and Electronic Engineers., pp. 651–654.

Rauch, H. E. (1963), 'Solutions to the linear smoothing problem.', *IEEE Transactions on Automatic Control* **8**, 371–372.

Renals, S. & Hochberg, M. (1995), Decoder technology for connectionist large vocabulary speech recognition, Technical Report +CS-95-17, Dept. of Computer Science, University of Sheffield. Dept. of Computer Science.

Renals, S. & Hochberg, M. (1999), 'Start-synchronous search for large vocabulary continuous speech recognition.', *IEEE Transactions on Speech and Audio Processing* **7**, 542–553.

Richards, H. & Bridle, J. S. (1999), The HDM: A segmental hidden dynamic model of coarticulation, *in* 'Proc. ICASSP', Phoenix, Arizona, USA.

Richardson, M., Bilmes, J. & Diorio, C. (2000*a*), Hidden-articulator Markov models for speech recognition, *in* 'Proc. ASR2000'.

Richardson, M., Bilmes, J. & Diorio, C. (2000*b*), Hidden-articulator Markov models: Performance improvements and robustness to noise, *in* 'Proc. ICSLP', Beijing, China.

Richmond, K. (2001), Estimating Articulatory Parameters from the Acoustic Speech Signal, PhD thesis, Centre for Speech Technology Research, Edinburgh University.

Richmond, K., King, S. & Taylor, P. (2003), 'Modelling the uncertainty in recovering articulation from acoustics', *Computer Speech and Language* **17**(2), 153–172.

Robinson, A. (1994), 'An application of recurrent nets to phone probability estimation', *IEEE Transactions on Neural Networks* **5**(2), 298–305.

Robinson, A., Cook, G., Ellis, D., Fosler-Lussier, E., Renals, S. & Williams, D. (2002), 'Connectionist speech recognition of broadcast news', *Speech Communication* **37**, 27–45.

Rosti, A.-V. & Gales, M. (2003), Switching linear dynamical systems for speech recognition, *in* 'UK Speech Meeting, London, April 2003', University College, London.

Rosti, A.-V. I. & Gales, M. J. F. (2001), Generalised linear Gaussian models, Technical Report CUED/F-INFENG/TR.420, Cambridge University Engineering.

Rosti, A.-V. I. & Gales, M. J. F. (2002), Factor analysed HMMs, *in* 'Proc. ICASSP'.

Roweis, S. (1999), Data Driven Production Models for Speech Processing, PhD thesis, California Institute of Technology, Pasadena, California.

Roweis, S. (2001), Personal communication.

Roweis, S. & Ghahramani, Z. (1999), 'A unifying review of linear Gaussian models.', *Neural Computation* **11**(2).

Russell, M. (1993), A segmental HMM for speech pattern modelling, *in* 'Proc. ICASSP', pp. 499–502.

Shumway, R. & Stoffer, D. (1982), 'An approach to time series smoothing and forecasting using the EM algorithm.', *Journal of Time Series Analysis* **3**(4), 253–64.

Siu, M., Iyer, R., Gish, H. & Quillen, C. (1998), Parametric trajectory mixtures for LVCSR, *in* 'Proc. International Conference on Spoken Language Processing'.

Smyth, P. (1998), 'Belief networks, hidden Markov models, and Markov random fields: a unifying view', *Pattern Recognition Letters.* .

Soong, F. & Huang, E. (1990), A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition, *in* 'Proc. Workshop on speech and natural language', Morgan Kaufmann Publishers Inc., pp. 12–19.

Stephenson, T., Bourlard, H., Bengio, S. & Morris, A. (2000), Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables, *in* 'Proc. ICSLP', Vol. II, pp. 951–954.

Stevens, K. (1999), *Acoustic Phonetics*, The MIT Press, Cambridge, Mass.

Stolcke, A. & Omohundro, S. (1992), Hidden Markov model induction by Bayesian model merging, *in* S. J. Hanson, J. D. Cowan & C. L. Giles, eds, 'Advances in Neural Information Processing Systems', Vol. 5, Morgan Kaufman.

Stolcke, A. & Omohundro, S. (1994), Best-first model merging for hidden Markov model induction, Technical Report TR-94-003, ICSI, Berkeley.

Sun, J., Jing, X. & Deng, L. (2000), Data-driven model construction for continuous speech recognition using overlapping articulatory features, *in* 'Proc. ICSLP'.

Taylor, P., Caley, R., Black, A. & King, S. (1997-2003), 'Edinburgh Speech Tools', `http://www.cstr.ed.ac.uk/projects/speech_tools`.

Verhasselt, J., Cremelie, N. & Marten, J. (1998), A hybrid segment-based system for phone and word recognition, *in* 'COST'.

Viterbi, A. J. (1967), 'Error bounds for convolutional codes and an asymptotically optimal decoding algorithm.', *IEEE Transactions on Information Processing* **13**, 260–269.

Westbury, J. (1994), *Ray Microbeam Speech Production Database User's Handbook*, University of Wisconsin, Madison, WI.

Williams, C. (2003), On representing the likelihood of an ar process in terms of difference observations. School of Informatics, University of Edinburgh.

Woodland, P. (1992), Hidden markov models using vector linear prediction and discriminative output distributions., *in* 'Proc. ICASSP', pp. 509–512.

Wrench, A. A. (2001), A new resource for production modelling in speech technology, *in* 'Proc. Workshop on Innovations in Speech Processing'.

Wrench, A. & Hardcastle, W. (2000), A multichannel articulatory speech database and its application for automatic speech recognition, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 305–308.

Wrench, A. & Richmond, K. (2000), Continuous speech recognition using articulatory data, *in* 'Proc. ICSLP', Beijing.

Young, S. (1993), The HTK hidden Markov model toolkit: Design and philosophy, Technical Report TR.153, Department of Engineering, Cambridge University.

Young, S. (1995), Large vocabulary continuous speech recognition: A review, *in* 'Proc. IEEE Workshop on Automatic Speech Recognition and Understanding', Snowbird, Utah, pp. 3–28.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. (2002), *The HTK Book (for HTK Version 3.2*, Cambridge University Engineering Department.

Young, S., Russell, N. & Thornton, J. (1989), Token passing: A simple conceptual model for connected speech recognition systems, Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept.

Yun, Y.-S. & Oh, Y.-H. (2002), 'A segmental-feature HMM for continuous speech recognition based on a parametric trajectory model', *Speech Communication* **38**(1-2), 115–130.

Zacks, J. & Thomas, T. (1994), 'A new neural network for articulatory speech recognition and its application to vowel identification', *Computer, Speech and Language* **8**, 189–20.

Zavaliagkos, G., Zhao, Y., Schwartz, R. & Makhoul, J. (1994), 'A hybrid segmental neural net/hidden Markov model system for continuous speech recognition', *IEEE Trans. on Speech and Audio Processing* **2**(1), II:151–160.

Zlokarnik, I. (1995*a*), 'Adding articulatory features to acoustic features for automatic speech recognition.', *Journal of the Acoustical Society of America* **97**(5 pt. 2).

Zlokarnik, I. (1995*b*), A speech recogniser using electromagnetic articulography. (English short version of PhD thesis).

Zweig, G. & Russell, S. (1998), Probabilistic modelling with Bayesian networks for automatic speech recognition., *in* 'Proc. ICSLP'.