# Full Covariance Modelling for Speech Recognition

Peter Bell

*Doctor of Philosophy*

School of Philosophy, Psychology and Language Sciences

The University of Edinburgh

2010

# Declaration

I hereby declare that this thesis is of my own composition, and that it contains no material previously submitted for the award of any other degree. The work reported in this thesis has been executed by myself, except where due acknowledgement is made in the text.

Peter Bell

May 24, 2010

Word count: 40,000

# Abstract

HMM-based systems for Automatic Speech Recognition typically model the acoustic features using mixtures of multivariate Gaussians. In this thesis, we consider the problem of learning a suitable covariance matrix for each Gaussian. A variety of schemes have been proposed for controlling the number of covariance parameters per Gaussian, and studies have shown that in general, the greater the number of parameters used in the models, the better the recognition performance. We therefore investigate systems with full covariance Gaussians. However, in this case, the obvious choice of parameters – given by the sample covariance matrix – leads to matrices that are poorly-conditioned, and do not generalise well to unseen test data. The problem is particularly acute when the amount of training data is limited.

We propose two solutions to this problem: firstly, we impose the requirement that each matrix should take the form of a Gaussian graphical model, and introduce a method for learning the parameters and the model structure simultaneously. Secondly, we explain how an alternative estimator, the shrinkage estimator, is preferable to the standard maximum likelihood estimator, and derive formulae for the optimal shrinkage intensity within the context of a Gaussian mixture model. We show how this relates to the use of a diagonal covariance smoothing prior.

We compare the effectiveness of these techniques to standard methods on a phone recognition task where the quantity of training data is artificially constrained. We then investigate the performance of the shrinkage estimator on a large-vocabulary conversational telephone speech recognition task.

Discriminative training techniques can be used to compensate for the invalidity of the model correctness assumption underpinning maximum likelihood estimation. On the large-vocabulary task, we use discriminative training of the full covariance models and diagonal priors to yield improved recognition performance.

# Acknowledgements

I'd like to thank Simon King, my supervisor, for all his advice, encouragement and enthusiasm. I'd also like to thank the Centre for Speech Technology Research for funding my PhD and providing a wonderful academic environment in which to work.

*A lengthier list of acknowledgements appears in hard copies of this thesis.*

# Contents

# Chapter 1

# Introduction

## 1.1 Data-driven automatic speech recognition

On its simplest level, Automatic Speech Recognition may be viewed as a pattern recognition problem (Bishop, 1995). In computer pattern recognition (specifically, classification) we are presented with an input associated with one of a discrete set of classes, and we wish to find some means of automatically determining the correct class. We seek a mapping from $\mathcal{X}$, the set of possible inputs, to $\mathcal{C} = \{C_1, C_2, \dots\}$, the set of possible outputs. This usually takes the form of a mathematical function, $h : \mathcal{X} \to \mathcal{C}$ In this context, $x$ is a a *feature vector*.

In practice, finding a good function $h$ can be extremely complex. In data-driven pattern recognition (also called *machine learning*), we split the problem into two parts: we write

$$h(x) \equiv h(x; \theta)$$

So that $h$ depends on some parameter $\theta$. The form of the function is specified using human knowledge, whilst $\theta$ is determined automatically from representative *training data*, a process known as *training* or *learning*. The training data may consist of sample pairs $(x_r, y_r)$ of inputs, together with the correct output classes, in which case the training is *supervised*, or simply example inputs $x_r$, in which case it is *unsupervised*.

Typically pattern recognition problems have inherent variability: a given output $C_k$ is not associated with a fixed input. It may even be that across all the data we find instances where an identical input is associated with multiple classes, so that it is not even possible to obtain a function capable of classifying all data correctly. This motivates *statistical* pattern recognition: here we model the variability explicitly, so that an input $x$ has a conditional probability $P(y|x)$ of being associated with class $y$. Classifying an input according to

$$\hat{y} = \arg\max_{y \in \mathcal{C}} P(y|x) \tag{1.1}$$

then gives the highest probability of the classification being correct. Of course, this probability is unknown. In data-driven statistical pattern recognition, we attempt to approximate it by a parametrised version, $P_\theta(y|x)$. Our classification function $h$ becomes

$$h(x) = \arg\max_{y \in \mathcal{C}} P_\theta(y|x) \tag{1.2}$$

as the pattern recognition problem is split into three parts: choosing the form of the parametrised probability $P_\theta(y|x)$; learning the optimal $\theta$ automatically from training data; and finally, given input data $x$, finding the class that maximises the probability. If we know the underlying frequencies of the respective classes, $P(y)$, we can use Bayes' theorem to rewrite the above equation as

$$h(x) = \arg\max_{y \in \mathcal{C}} p_\theta(x|y)P(y) \tag{1.3}$$

Almost all modern speech recognition systems are based on these techniques. In building a speech recognition system, we construct a mathematical model that takes a recorded speech utterance as its input and returns a natural language transcription as its output. To do this, we must specify the form of the probability model, then train its parameters using some transcribed training utterances. When presented with new utterances, we then compute the most likely transcription, using the trained model.

Speech recognition is a hard machine learning problem:

- The input space $\mathcal{X}$ is high-dimensional: digital recordings of speech commonly sample the speech at frequencies of 16khz. We might extract somewhere between 12 and 60 continuous frequency coefficients for each 10ms interval , so that a 10s utterance could be associated with a 60,000-dimensional input space.

- The output space, too, is high-dimensional. A large-vocabulary system might have a vocabulary of the order of hundreds of thousands of words.

- Speech is inherently dynamic. The feature vector is not fixed in length: it depends on the length of the utterances. The output is not a single word: it is a string of words of unknown length. Moreover, it is unknown which parts of the feature vector correspond to which word, or even which part of a word.

- Speech is highly variable. It varies between groups of speakers due to differences in accent, and between individuals due to physical differences in speech production, most obviously due to differences in age and sex. It varies with speaking rate and speaking style.

- In real-world situations, recorded speech data may be degraded due to background noise or reverberation. Additionally, the data may be degraded due to channel conditions, for example, when the signal has been transmitted by telephone.

Humans find speech recognition difficult too. When the speaker has an unfamiliar accent, or is speaking in a crowded room, or over the telephone, we often struggle to hear what is being said. In these situations we rely on strong intuitions about what is likely to be said, based on our knowledge of language, the speaker, and the situation. When we are unable to rely on this mental model – for example, when recognising strings of isolated alphanumeric characters, such as postcodes, about which we have no prior knowledge – we often fail to recognise speech correctly. In designing a statistical ASR system, then, we require a model which:

1. can deal with the dynamic, high-dimensional nature of speech and written language;

2. uses the knowledge we have about human speech perception;

3. uses a powerful statistical model capable of modelling the variability in speech, whose parameters we can learn from available data

Early ASR systems used dynamic time warping (Sakoe & Chiba, 1978; Vintsyuk, 1968) to force two utterances to be compared to have the same length. However, since the early 90s, Hidden Markov Models (HMMs) have become predominant. Introduced by a team at the Institute for Defense Analyses (Ferguson, 1980) and in the Dragon system by Baker (1975), and developed extensively by the IBM speech research group (Bahl *et al.*, 1983), HMMs essentially reduce the dynamic training and recognition problems to a series of static inference and classification problems, one for each frame. They do this by making simple assumptions about the conditional independence of successive frames of speech, given some hidden variable.

Human knowledge plays a role in system design: in constructing the feature vector $x$ from the frames of speech, we employ front-end processing that is typically motivated by speech production or perception. Linear prediction coefficients (Markel & Gray, 1976, for example) arise from modelling the human vocal tract as an all-pole filter. The Mel-filterbank (Stevens & Newman, 1937) models the frequency response of the ear. Perceptual linear prediction (Hermansky, 1990) combines the two. Motivated by linguistic theories, we model words as sequences of discrete, perceptually categorical speech sounds, known as phones. To substitute for the role of high-level domain knowledge in human speech recognition, we use a prior model for the lexical content of speech, called a *language model*, independent of the speech acoustics. N-gram language models, described in Chapter 2, are powerful and can be conveniently integrated into the HMM-decoding process.

In an HMM system, acoustic modelling is concerned with finding the probability $p_\theta(x|y)$ - (from Equation 1.3) for the feature vector $x$ for each frame. In this context $y$ represents some categorical acoustic variable, such as a phone. In early HMM systems (Shikano, 1982), the probability was obtained via a discretisation of the acoustic space using a process known as Vector Quantisation (VQ). Using a clustering algorithm, a set of codebook vectors is obtained, and the acoustic space is then partitioned into segments according to which codebook vector is the closest. Rabiner *et al.* (1985) first used parametric continuous probability distributions with HMMs for acoustic modelling, modelling $p_\theta(x|y)$ using mixtures of multivariate Gaussians (GMMs). Probably the most widely used acoustic models used in ASR systems, GMMs are flexible and powerful, and we use them for all systems described in this thesis. We refer to the HMM-GMM combination as a continuous-density HMM (CD-HMM). The density of each Gaussian $m$ is parametrised by its mean $\mu_m$ and covariance matrix $\Sigma_m$.

Humans learn speech recognition from a limited set of speakers, but we possess a capacity to generalise what we learn to new speakers and speaking conditions with minimal new examples, in a way that is unmatched by any automatic system. However, we have very little understanding of the learning mechanism used by the brain, and efforts to replicate it have met with limited success. The most effective alternative is to use complex models with very high modelling power, and train them using very large amounts of data. CD-HMMs are very suitable for this purpose: from an engineering perspective, they have the attraction that there are well-understood and computationally tractable algorithms for parameter inference and probability maximisation using these models. We describe them in Chapter 2. The complexity of the models may easily be controlled by several different means: controlling the cardinality of $y$; varying the number of Gaussians in the mixture model; and varying the number of free parameters of each Gaussian.

We could imagine that, were we supplied with a near infinite quantity of training data, covering every possible set of speaker characteristics, accent, speaking style and environmental condition, and given sufficient computational resources, we could build an CD-HMM system capable of recognising speech as well as a human – or at least as well as a human listening to speech of unfamiliar content, without strong linguistic prior knowledge. Certainly systems perform well enough when the input is from a single speaker with prescribed conditions, and there is plenty of representative training data. In practice, the limited availability of training data, relative to the range of speech that the system is required to process, poses a major problem. If the system is complex enough to deal with all potential input speech, it will be prone to over-fitting to the specific training data available and will perform poorly on new data, whilst if it is too simple, it will lack the power to model the full range of expected input – this is illustrated in Figure 1.1. It is usually necessary to seek a trade-off between the two extremes. The problem is one of generalisation. It is essential to have a system that is tuned so that, given the quantity of training data available, it generalises as well as possible to unseen data.

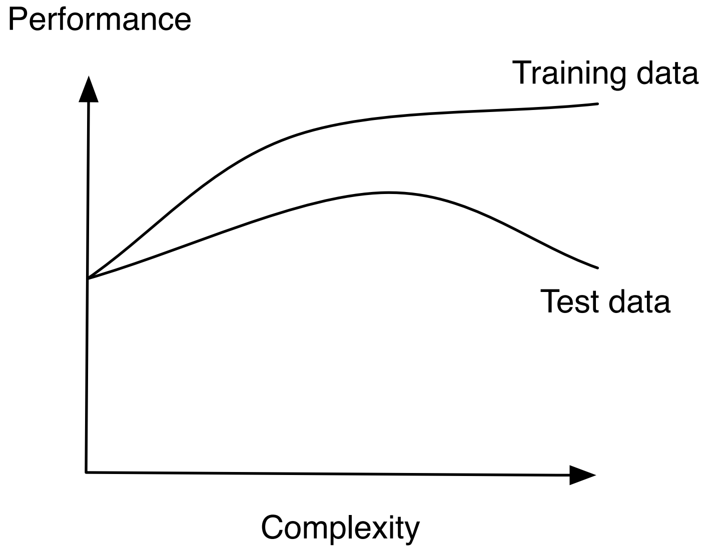

Figure 1.1: An illustration of the problem of model generalisation: when the model is too simple, it generalises to unseen test data, but lacks the modelling power necessary for high performance. When the model is too complex, it overfits to the data used to train it, performing well on this data, but generalising poorly to unseen data.

In this thesis we assume that training data is always limited. We investigate methods for maintaining generalisation ability when very high-complexity models are used. We focus specifically on the estimation of the covariance matrices, $\Sigma_m$, used to compute the Gaussian densities, which are used in turn to compute $p_\theta(x|y)$. In their unconstrained form the covariance matrices have very large numbers of free parameters, so the issue of generalisation versus modelling power is particularly pertinent.

We are particularly interested in developing methods to maintain generalisation that are universally applicable, without requiring extensive task-dependent hand tuning. Our goal is to automatically adjust the model for optimal performance given the available training data using the knowledge with which we are naturally provided about its quantity and variability. Ultimately, we seek methods which, when averaged over all the training sets we might encounter, result in $p_\theta(x|y)$ being as close as possible to the real, model-free $p(x|y)$ (or at least, for the classification decisions made using the respective probabilities to be close). That requires not only models that generalise well, but also models that compensate the limitations of our CD-HMM parametrisation.

## 1.2 Contribution

Having presented the broad motivations of this thesis, we now outline the process which lead to these goals; we briefly summarise the novel research undertaken.

Work by Bilmes (1999) on Buried Markov Models and continuous variable Dynamic Bayesian Networks lead us to begin study of full covariance Gaussian systems. In a Buried Markov Model, the dependency structure between elements of the acoustic feature vector varies with the hidden state. This is equivalent to imposing a sparsity structure on the covariance parameters. When the dependency structure is specified using an undirected graphical model, this corresponds to the sparsity structure of the inverse covariance matrix. This was first studied by Dempster (1972), and can be viewed within a more general class of precision matrix models (Sim & Gales, 2004).

Working within a generative modelling framework, we first investigated the problem of learning the sparse dependency structures directly from data, with the aim of finding a globally optimal solution. This lead us to the recently-developed optimisation method of Banerjee *et al.* (2006). Their method is related to the lasso (Meinshausen & Bühlman, 2006), and requires an $l_1$ penalty term to be included when maximising the likelihood. This work motivated a more general consideration of the desirable properties of a covariance estimator, leading us to investigate, as an alternative, a shrinkage estimator (Stein, 1956). Here, the full matrix is interpolated with a lower dimensionality estimator to optimise the trade-off between variance and bias.

We implemented the two estimation techniques for training the parameters in an CD-HMM system, and compared their performance for phone recognition using the TIMIT corpus, artificially constraining the training data available. We found that the performance of the shrinkage estimator was significantly higher than that of the lasso estimator. We went on to investigate the former's performance on a large-vocabulary conversational telephone speech recognition task. We compared it with smoothing methods described slightly earlier by Povey & Saon (2006). We carried out a detailed analysis of the statistics used by the shrinkage estimator to relate the two techniques.

The generative framework has been shown to be sub-optimal for ASR due to its reliance on the assumption of model correctness. This can be remedied by explicitly discriminative parameter estimation. We therefore integrated discriminative training into the full covariance techniques. We describe the recipes used for training based on the MMI criterion (Bahl *et al.*, 1986).

This thesis includes a thorough review of the work of others in the field of covariance modelling. We also describe other work that has been used to improve ASR performance, where we have incorporated that work in our systems. The style of prose used, and particularly the fact that we have attempted to express the work of others using a consistent framework throughout the document, occasionally makes it difficult to discriminate between existing research and new research carried out for this thesis. We therefore briefly list the original contributions:

- the use of $l_1$-penalised likelihood for learning sparse precision matrix structure within a Gaussian mixture model framework for ASR; experiments varying the penalty parameter on an ASR task;

- a comparison of the effect of varying the quantity training data on the performance of different covariance models for phone recognition;
- an in-depth analysis of the effect of off-diagonal smoothing of full covariance models on both phone recognition and large vocabulary ASR tasks;
- a mathematical comparison of Bayesian and classical methods for covariance estimation; the derivation of a formula for estimating shrinkage parameters for an CD-HMM system from data, and a method for sharing the required statistics across Gaussians;
- a comparison of the effects of covariance smoothing when models are generatively and discriminatively trained.

In addition, we have tried to make accessible some literature from outside the field of ASR – most notably that related to convex optimisation for $l_1$-penalised likelihood maximisation and consistency of covariance estimation under weak asymptotic assumptions. To this end, the thesis contains substantial mathematical reformulation of the original work, so that whilst the results are not new, several of the derivations are.

The following publications are almost entirely composed of work contained in this thesis:

Bell, P. & King, S. (2007). Sparse Gaussian graphical models for speech recognition. In *Proc. Interspeech*.

Bell, P. & King, S. (2008). A shrinkage estimator for speech recognition with full covariance HMMs. In *Proc. Interspeech*.

Bell, P. & King, S. (2009). Diagonal priors for full covariance speech recognition. In *Proc. ASRU*.

## 1.3 Structure

The remainder of the the thesis is structured as follows:

- In Chapter 2 we describe the main components of a CD-HMM speech recognition system. We introduce the main algorithms used for HMM training and decoding and discuss the common system refinements.

- Chapter 3 introduces covariance modelling. We consider the desired properties of a covariance model, and describe parameter estimation. We discuss various approaches to covariance modelling previously employed in ASR systems.

- Chapter 4 gives background on graphical modelling and relates work in this area to covariance modelling. We discuss structure learning in graphical models, and describe the convex optimisation methods used in this thesis.

- Chapter 5 describes the shrinkage estimator and its properties. We derive formulae for the optimal shrinkage parameter, and compare this to other covariance smoothing methods.

- In Chapter 6 we present results on the TIMIT phone recognition task using the two techniques for full covariance estimation, when the amount of available training data is constrained. We then describe our large vocabulary system, and present a range of full covariance results.

- Chapter 7 explains the need for discriminative training, gives background on the techniques used. We derive formulae for full covariance discriminative training, and considers covariance regularisation within a discriminative framework.

- In Chapter 8 we describe our recipes for training discriminative full covariance models for the large vocabulary recognition system, and present results.

- Chapter 9 summarises the work and discusses potential future research.

## 1.4 Notation

When choosing mathematical notation, we have tried to strike a balance between maintaining consistency throughout the thesis and retaining consistency with original references or standard conventions in the literature. Inevitably, the latter concern means that notational conflicts occur: for example, $\beta$ is used to denote HMM forward probabilities, but also sums of state posteriors. In these cases, the use should be clear from the context. For quantities recurring throughout this document, however, we have tried to use standard symbols, even when these may differ from those used in the original references.

Some general conventions that we adopt are: vectors and scalars are not differentiated, since usually the difference is unimportant; matrices are denoted by upper case letters; and for statistical parameters, we use Greek letters for unknown parameters and Roman letters for estimates obtained from training data and for random variables. Overleaf is a description of recurrently-used symbols.

| | |
|---|---|
| $O$ | Sequence of observation vectors |
| $o_t$ | Observation at time $t$ |
| $Q$ | Sequence of hidden HMM states |
| $q_t$ | HMM hidden state at time $t$ |
| $W$ | Sequence of words |
| $\gamma$ | State/Gaussian occupation probability |
| $\beta$ | Sum of state posteriors |
| | |
| $m$ | Index over Gaussians |
| $r$ | Index over utterances |
| $i, j$ | Indices over matrix/vector elements (usually) |
| $t$ | Index over time |
| | |
| $A$ | Arbitrary matrix |
| $R$ | Arbitrary rotation matrix |
| $T$ | As superscript, matrix/vector transpose; otherwise, total time frames |
| $\Sigma$ | Unknown "true" covariance matrix |
| $S$ | Sample covariance matrix |
| $U$ | Covariance matrix learned from data, constrained by some model |
| $P$ | Precision matrix learned from data |
| $D$ | Diagonal matrix (NB. also has other uses) |
| $\Lambda$ | Diagonal matrix of eigenvalues |
| $\lambda$ | Eigenvalue of a symmetric matrix |
| | |
| $\tau$ | Smoothing prior weight |
| $\alpha$ | Shrinkage parameter |
| $\rho$ | Likelihood penalty parameter |
| $x$ | Arbitrary sample vector |
| $y$ | Class label |
| $n$ | Number of samples |
| $d$ | Dimensionality |
| $\theta$ | Arbitrary set of parameters |

# Chapter 2

# Speech recognition with continuous-density HMMs

Applying front end feature processing to a recorded utterance, we obtain a series of $d$-dimensional observation vectors $O = (o_1, o_2, \ldots, o_T)$, where $T$ is the number of frames. The goal of an ASR system is to predict the correct word sequence $W$ for the utterance. We aim to find the most likely word sequence, given by

$$\hat{W} = \arg\max_W P(W|O) \tag{2.1}$$

Following 1.3, we can use Bayes' theorem to write this as

$$\hat{W} = \arg\max_W p(O|W)P(W) \tag{2.2}$$

The problem of finding $P(W)$ – which is independent of the recorded utterance – is termed language modelling. We discuss this briefly in Section 2.2.3. Finding $p(O|W)$ is termed acoustic modelling. We now explain how this may be approximated using a continuous-density hidden Markov model.

In practice, we compute log probabilities to avoid numerical underflow, and scale the language model log probabilities by a factor $\nu$ to compensate for the fact the the acoustic models lead to a distribution that is narrower than the true distribution, due to the conditional independence assumptions made. Thus, we seek the maximum of

$$\log p(O|W) + \nu \log P(W) \tag{2.3}$$

In this chapter we describe the standard features of an ASR system using continuous-density hidden Markov models. We focus particularly on acoustic modelling. We first introduce the HMM and describe algorithms used for parameter training and decoding, and describe practical issues of model construction. We then describe the Gaussian mixture model, used as the probability density function for the acoustic space. Finally, we discuss some commonly-used refinements for acoustic modelling.

## 2.1 The hidden Markov model

### 2.1.1 Model assumptions

The hidden Markov model (HMM) models $p(O|W)$ via an intermediate sequence of discrete hidden (i.e. unobservable) variables $Q = (q_1, \dots \dots, q_T)$, one for each frame $t$. $Q$ is called the *state sequence*. We assume that the value of the hidden variable $q_t$ exclusively determines the acoustic properties of that frame. In other words, we assume that an observation $o_t$ is conditionally independent of all other observations, and other hidden states, given $q_t$:

$$p(o_t|O, Q) = p(o_t|q_t) \tag{2.4}$$

These are known as the HMM observation or "output" probabilities, and we say that the hidden state "generates" the observation.

We also make a first-order Markov assumption: that given the preceding state, each state is conditionally independent of all earlier states

$$P(q_t|q_1, \dots q_{t-1}) = P(q_t|q_{t-1}) \tag{2.5}$$

The state sequence is therefore a Markov chain (Grimmet & Stirzaker, 1982). These assumptions are illustrated in the Graphical Model in Figure 2.1. Using the assumptions, the probability of the observed acoustics is computed as:

$$p(O|W) = \prod_{t=1}^{T} P(q_t|q_{t-1})p(o_t|q_t) \tag{2.6}$$

$q_0$ may be set as some fixed entry state. The word transcription $W$ is used to constrain the model topology. Indexing the set of possible states by $j$, we write the output probabilities as

$$b_j(o_t) := p(o_t|q_t = j) \tag{2.7}$$

and also the transition probabilities

$$a_{ij} := P(q_t = j|q_{t-1} = i) \tag{2.8}$$

We use $\theta$ as a shorthand to refer to all the parameters of the HMM, and occasionally use $p_\theta(O)$, $p_\theta(O|W)$, etc to denote that the probabilities are computed with HMM assumptions and parametrisation, contrasting with the true underlying probabilities.

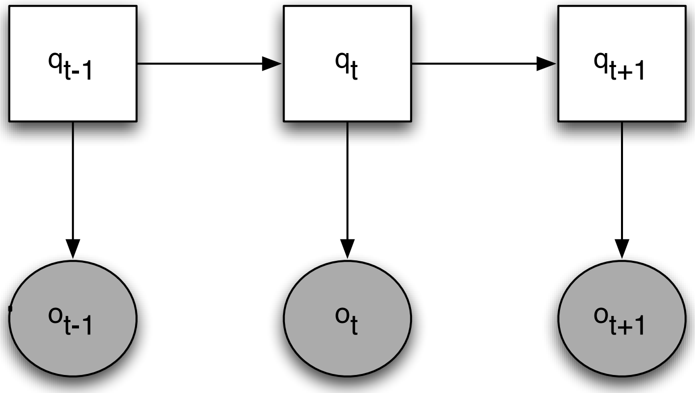

Figure 2.1: A graphical model illustrating the dependencies in an HMM. The lack of an arrow between two variables indicates that they are conditionally independent.

### 2.1.2 HMM topology

To relate the state sequence $Q$ to the word sequence $W$, it is necessary to define the state sequence topology. We take words to be strings of sub-word units, say phones. Ignoring pronunciation variation, the mapping is deterministic. Most systems model phones as HMMs with three distinct "emitting states" (states associated with an observation), each with their own output probability functions. By convention these are labelled from 2-4, to allow for entrance and exit states, which simplifies the topology. In the Markov chain, transitions can occur from each emitting state to the next, and from each emitting state to itself. In this

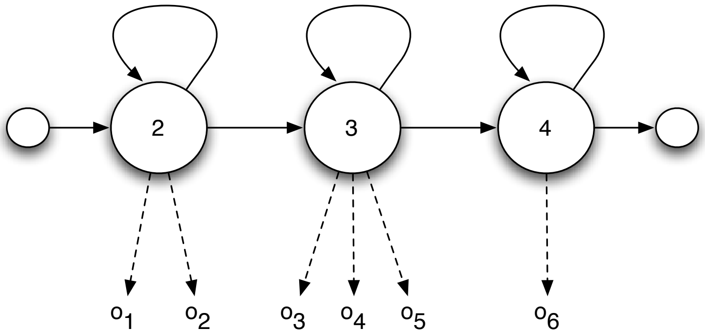

Figure 2.2: The topology of an HMM with three emitting states. In this example there are six observations.

way, each phone state can "generate" multiple successive observations. This is illustrated in Figure 2.2

Ignoring pronunciation variation, the sequence of phone states is deterministically related to the word sequence $W$. This is specified by a lexicon. During training, when the word transcription is supplied, the model topology for the utterance is fully specified. During decoding, the probability of transitions between words is supplied by the language model. Figure 2.3 illustrates a simple network for word recognition with a small vocabulary.

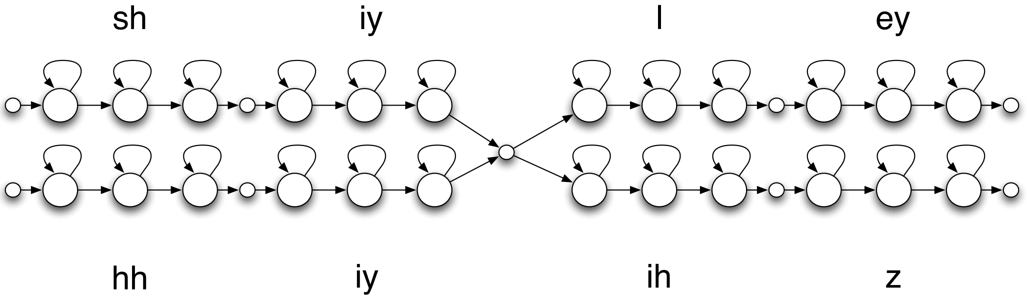

Figure 2.3: A network of HMMs for recognition of (she/he)(lay/is). The word models are constructed from three-state phone models, each with three emitting states. Inter-word phone transitions are deterministic.

### 2.1.3 The forward-backward algorithm

To update the HMM parameters during training, it is necessary to compute $P(q_t|O, W, \theta)$, where $\theta$ is some initial parameter set. This may be computed efficiently using the forward-backward algorithm (Rabiner & Juang, 1993). We define the forward probabilities

$$\alpha_t(j) = p(o_1, \ldots, o_t, q_t = j|\theta, W) \tag{2.9}$$

and backward probabilities

$$\beta_t(j) = p(o_{t+1}, \ldots, o_T|q_t = j, \theta, W) \tag{2.10}$$

Then we have

$$p(O, q_t = j|\theta, W) = \alpha_t(j)\beta_t(j) \tag{2.11}$$

$$p(O|W) = \sum_i \alpha_t(i)\beta_t(i) \tag{2.12}$$

with the latter holding for any $t$. Then

$$P(q_t = j|O, \theta, W) = \frac{p(O, q_t = j|\theta, W)}{p(O|\theta, W)} = \frac{\alpha_t(j)\beta_t(j)}{\sum_i \alpha_t(i)\beta_t(i)} \tag{2.13}$$

We denote this by $\gamma_j(t)$ and refer to it as a state posterior probability. The forward and backward probabilities are computed inductively:

$$\alpha_{t+1}(j) = \sum_i \alpha_t(i)a_{ij}b_j(o_{t+1}) \tag{2.14}$$

$$\beta_t(i) = \sum_j a_{ij}b_j(o_{t+1})\beta_{t+1}(j) \tag{2.15}$$

The use of the forward-backward probabilities for HMM parameter estimation using the Expectation-Maximisation algorithm is known as the Baum-Welch algorithm (Baum *et al.*, 1970), and is discussed in Section 2.3.2.

### 2.1.4 The Viterbi algorithm

The Viterbi algorithm is used to find the hidden state sequence giving the highest likelihood to the observations:

$$\hat{Q} = \arg\max_Q p(Q, O|\theta) \tag{2.16}$$

We define

$$\phi_j(t) = \max_Q p(o_1, \ldots, o_t, q_t = j|\theta) \tag{2.17}$$

where the maximum is taken over all paths ending in state $j$ at time $t$. These probabilities can be computed inductively:

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\} b_j(o_t) \tag{2.18}$$

At each time we store, for every $j$, the probability $\phi_j(t)$ and the identity of the previous state $i$ from which the probability was obtained. This allows the complete most likely sequence to be recovered by back-tracing after the iterations are complete. The use of the Viterbi algorithm for decoding is discussed further in Section 2.2.4.

## 2.2 Components of an HMM-based ASR system

We briefly describe the main components of an HMM-based ASR system. This is to facilitate the description of our experimental systems in later chapters; a more complete introduction may be found in Young (2008).

### 2.2.1 Acoustic feature extraction

The acoustic front-end processes the raw speech waveform to extract the acoustic features $o_t$ for use in the HMM. The aim of the process is to obtain features that are useful for phone discrimination, whilst removing those conveying non-lexical information such as emphasis and emotion. The feature extraction should also minimise the effect of variation due to speaker and recording conditions. The features should not be strongly correlated to avoid redundant model parameters.

The two most commonly used features are Mel Frequency Cepstral Coefficients (MFCCs) (Davis & Mermelstein, 1980) and Perceptual Linear Prediction (PLPs) (Hermansky, 1990). The speech is transformed to the frequency domain using a Fourier transform with a Hamming window. In the HTK implementation we used, a Mel-scale filterbank is then applied. This attempts to replicate the frequency response of the human ear. To obtain PLPs we estimate the coefficients of an all-pole filter modelling the vocal tract transfer function. In both cases coefficients are converted into the cepstral domain, and a discrete cosine transform is applied. For more details see Young *et al.* (2006). Using cepstral domain features has the advantage that some of the channel effects can be removed by normalising the mean of the coefficients. This is known as Cepstral Mean Normalisation (CMN). The mean is usually estimated on a per-recording basis. Similarly, Cepstral Variance Normalisation (CVN) can be applied.

The feature vectors are usually appended with the coefficient differentials, second differentials, and possibly third differentials. This helps compensate for the invalidity of the assumption that successive observations are conditionally independent without adding undue complexity.

### 2.2.2 Sub-word units

In all but a very small vocabulary recognition system, it is necessary to identify sub-word units to define the HMM states. Phones – perceptually distinct speech sounds, of which there are 40-50 in English – are a natural choice of unit. However, the acoustic realisation of a phone is strongly dependent on the context, particularly in faster, spontaneous speech, due to co-articulation. To solve this problem, context-dependent phone units are used. Triphones are composed of a central phone, with one adjacent phone of context on either side. Context is included across word boundaries.

```
sil sh iy ih z hh ae p iy sil
sil sil-sh+iy sh-iy+ih ih-ih+z ih-z+hh z-hh+ae hh-ae+p ...
...ae-p+iy p-iy+sil sil
```

Figure 2.4: Monophone form for *she is happy* converted to cross-word triphone representation

The number of possible triphones is very large. Although many of them will not occur in speech at all, a problem arises when triphones that do occur naturally are not observed in the training data; and more generally, rarely seen triphones will have insufficient data for reliable parameter estimation. To avoid this, similar triphones are tied, sharing HMM states. Following Young *et al.* (1994) the tying is accomplished using binary decision trees with phonologically-based questions at each node.

### 2.2.3 Language modelling

Language modelling is the process of estimating the prior probability of a string of words, $W = (w_1, \ldots w_K)$. This is the $P(W)$ from Equation 2.2. An N-gram model makes the simplifying assumption that words are conditionally independent, given the $N - 1$ previous words, so that

$$P(W) = \prod_{k}^{K} P(w_k | w_{k-1}, \ldots w_{k-N+1}) \tag{2.19}$$

Given sufficiently high $N$ (say 3 or 4), these simple models are surprisingly good at modelling language. The probabilities are estimated using word co-occurrence counts in training data, $C(w_{k-N+1}, \ldots, w_k)$. These counts, however, are frequently close to zero for higher $N$, resulting in estimates that do not generalise well. One of the most successful solutions to this problem is modified Kneser-Ney smoothing (Chen & Goodman, 1999; Ney *et al.*, 1994), in which the language model probabilities are obtained using an interpolation from different order N-grams, with greater weight given to the lower order N-grams when data is sparse.

### 2.2.4 Decoding

The Viterbi algorithm, described in Section 2.1.4 is commonly used as a basis for decoder implementations. Some refinements are required to avoid the high computational cost due to the large search space in large-vocabulary decoding. *Beam search* removes tokens that deviate in likelihood by more than some fixed amount (the *beam width*) from the most likely hypothesis at each frame $t$. The beam width should not be too small, to prevent the most likely path being missed. The recognition network can be tree-structured so that computation (before pruning is effective) is shared for words sharing the first few phones.

Instead of producing a single output hypothesis, it is possible for the decoder to store multiple tokens at each frame to produce a *lattice*, efficiently encoding several of the most likely hypotheses (Richardson *et al.*, 1995; Thompson, 1990). In the standard representation, nodes in the lattice represent word start and end times, and arcs represent words, to which acoustic and language model probabilities can be attached.

Lattices can be used to efficiently apply a more powerful language model, such as a trigram or 4-gram. Known as *rescoring*, the arcs are updated with the new language model probabilities, and the new one-best hypothesis obtained. In addition, lattices may be used to specify the network for a second pass of decoding. This is particularly useful when the new acoustic models have a much greater computational costs. In both cases, it is important to ensure that the lattices are sufficiently large to ensure that some of the most accurate transcriptions are contained in the lattice.

## 2.3 Observation probabilities: the Gaussian mixture model

In this section we introduce the Gaussian mixture model (GMM) and explain how its parameters may be estimated using the EM algorithm.

### 2.3.1 The model

We model the output probabilities $b_j(o_t) = p(o_t|q_t = j)$ using a Gaussian mixture model. Given the state $j$, this assumes that the observation has a probability $c_{jm}$ of being generated by a multivariate Gaussian distribution with density function $f_{jm}(o_t)$

$$p(o_t|q_t = j) = \sum_{m}^{M_j} c_{jm} f_{jm}(o_t) \tag{2.20}$$

The $c_{jm}$ are also referred to as the mixture weights. To simplify the notation, where appropriate we drop the dependence on $j$, and use $m$ as an index to the global collection of Gaussians and weight parameters; note, however, that in our systems, as in standard CD-HMMs, Gaussians are shared only between states that have been explicitly tied.

The Gaussian probability density function $f_m(o_t)$ is defined as

$$f_m(o_t) = f(o_t\ ;\ \mu_m, \Sigma_m) = (2\pi)^{-d/2}|\Sigma_m|^{-1/2} \exp\{-\frac{1}{2}(o_t - \mu_m)^T \Sigma_m^{-1}(o_t - \mu_m)\} \tag{2.21}$$

where $\mu_m$ and $\Sigma_m$ are the mean and covariance parameters respectively. The Gaussian distribution has a number of desirable properties. It is a member of the exponential family of distributions. The variable $o_t$ appears only in the exponential term, which can be written

$$-\frac{1}{2}(o_t-\mu)^T\Sigma^{-1}(o_t-\mu) = -\frac{1}{2}\operatorname{tr}\Sigma^{-1}(o_t-\mu)(o_t-\mu)^T \tag{2.22}$$

$$= \operatorname{tr}(\Sigma^{-1}. -\frac{1}{2}o_t o_t^T) + \Sigma^{-1}\mu o_t^T - \frac{1}{2}\Sigma^{-1}\mu\mu^T \tag{2.23}$$

so we see that the density function can be written in canonical exponential form, with parameters $(\Sigma^{-1}\mu, \Sigma^{-1})$ and statistics $(o_t^T, -\frac{1}{2}o_t o_t^T)$. It is well known that the exponential distribution with canonical statistics $T(x)$ is the maximum entropy distribution, given $T(x)$. The Gaussian is therefore the maximum entropy distribution with fixed first and second order statistics. This is the distribution with the highest degree of uncertainty, given the data. We return to this theme in Chapter 4.

Unlike a standard Gaussian distribution (regardless of the number of parameters used), the mixture model is of course capable of modelling skewed and multimodal distributions; it is readily able to model data that forms distinct clusters in acoustic space. The complexity of the model can most easily be controlled by varying the number of Gaussians in the model.

### 2.3.2 Parameter estimation with the EM algorithm

In this section we describe the method for estimating the parameters of the Gaussian mixture model using the forward-backward probabilities introduced in Section 2.1.3. We do not discuss the estimation of other HMM parameters here (see Rabiner & Juang, 1993).

For now we adopt the standard approach, and assume that we wish to find parameters $\theta$ to maximise the likelihood of the training data. Given a training utterance with observations $O$ and transcription $W$, we attempt to maximise the log-likelihood:

$$F_{\mathrm{ML}}(\theta) = \log p(O|\theta, W) \tag{2.24}$$

No analytic solution exists for this maximisation. However, given some initial parameter set, it is possible to find an *auxiliary function* such that adjusting $\theta$ to increase the auxiliary function guarantees an increase in the objective function (2.24). For notational clarity we only consider one utterance here. However, the summations derived below easily extend over an entire collection of training utterances.

The procedure we describe is known as the Expectation-Maximisation (EM) algorithm (Dempster *et al.*, 1977). Its application to HMM parameter estimation using the forward-backward probabilities is known as the Baum-Welch algorithm (Baum *et al.*, 1970). Suppose that we have an initial parameter set $\theta^0$. We first use this parameter set to compute the joint posterior probability of a state sequence $Q$, and sequence of emitting Gaussians $M$, $P(Q, M|O, \theta^0, W)$. This is known as the E-step. We define the auxiliary function at $\theta^0$ by

$$G(\theta, \theta^0) := \sum_Q \sum_M P(Q, M|O, \theta^0, W) \log p(O, Q, M|\theta, W) \tag{2.25}$$

The increase in the log likelihood is given by:

$$F_{\text{ML}}(\theta) - F_{\text{ML}}(\theta^0) = \log p(O|\theta, W) - \log p(O|\theta^0, W) \tag{2.26}$$

By conditioning on $Q$ and $M$, we can express this increase as

$$\log \Big[ \sum_Q \sum_M P(Q, M|O, \theta^0, W) \frac{p(O, Q, M|\theta, W)}{P(Q, M|O, \theta^0, W)} \Big] - \log p(O|\theta^0, W) \tag{2.27}$$

$$\geq \sum_Q \sum_M P(Q, M|O, \theta^0, W) \log \Big( \frac{p(O, Q, M|\theta, W)}{P(Q, M|O, \theta^0, W)} \Big) - \log p(O|\theta^0, W) \tag{2.28}$$

$$= \sum_Q \sum_M P(Q, M|O, \theta^0, W) \Big[ \log p(O, Q, M|\theta, W) - \log P(Q, M|O, \theta^0, W) p(O|\theta^0, W) \Big] \tag{2.29}$$

$$= \sum_Q \sum_M P(Q, M|O, \theta^0, W) \Big[ \log p(O, Q, M|\theta, W) - \log p(O, Q, M|\theta^0, W) \Big] \tag{2.30}$$

$$= G(\theta, \theta^0) - G(\theta^0, \theta^0) \tag{2.31}$$

where the first step uses Jensen's inequality, and (2.29) uses the fact that

$$\sum_Q \sum_M \log P(Q, M|O, \theta^0, W) = 1 \tag{2.32}$$

So at each step, an increase in $G(\theta, \theta^0)$ is guaranteed to increase the log likelihood. We attempt to find new parameters to maximise this function – this is known as the M-step, a process we explain in more detail below. By repeating this procedure iteratively, we will find at least a local maximum of the objective function. This is illustrated in Figure 2.5. Note that the gradients of the objective function $F(\theta)$ and the auxiliary $G(\theta, \theta^0)$ are equal at $\theta^0$.

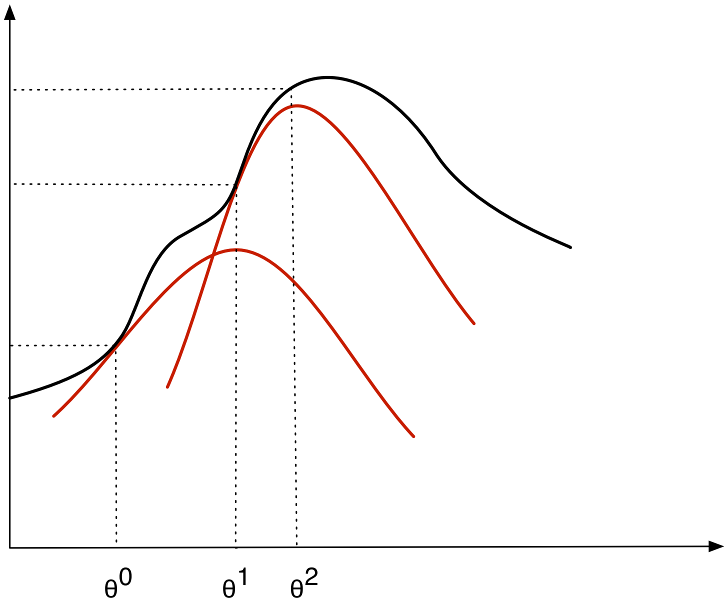

Figure 2.5: An illustration of two iterations of the EM algorithm. The objective function is shown in black. The auxiliary functions at $\theta^0$ and $\theta^1$ are shown in red. The horizontal axis represents parameter space; the vertical axis represents the value of the objective and auxiliary functions.

To use $G(\theta, \theta^0)$ to update the GMM parameters, we reformulate the sums over the sequences $Q$ and $M$ by a sum over the frames $t$, denoting posterior probability of Gaussian $m$ and state $j$ at time $t$ by

$$\gamma_{jm}(t) = P(m, q_t = j | O, \theta^0, W) \tag{2.33}$$

This may be computed using the forward-backward algorithm (see Equation 2.13). Ignoring the transition probability terms, which are not required for this discussion, the log probability $\log p(O, Q, M | \theta, W)$ may be factorised as a sum of log-probabilities $\sum_t \log p(o_t, m | \theta, W)$, since $o_t$ is dependent only on the state $q_t$, and we have

$$G(\theta, \theta^0) = \sum_t \sum_j \sum_m \gamma_{jm}(t) \log\left[p(o_t|m)P(m|q_t = j)\right] \tag{2.34}$$

$$= \sum_t \sum_j \sum_m \gamma_{jm}(t)(\log f_{jm}(o_t) + \log c_{jm}) \tag{2.35}$$

To update the parameters pertaining to the Gaussian $jm$, therefore, we need only maximise

$$\sum_t \gamma_{jm}(t)(\log f_{jm}(o_t) + \log c_{jm}) \tag{2.36}$$

$$= \sum_t \gamma_{jm}(t)[-\frac{1}{2}\log|\Sigma_{jm}| - \frac{1}{2}(o_t - \mu_{jm})^T \Sigma_{jm}^{-1}(o_t - \mu_{jm})] \tag{2.37}$$

As mentioned earlier, we usually drop the dependence on $j$.

For initialisation, it is common to set all parameters to the global mean and variance (known as a “flat start”). Alternatively, an earlier model set may be used, if it exists. For example, when training triphone models, parameters from from the corresponding monophones may be used.

### 2.3.3 Fitting the mixtures

GMMs are usually initialised with a single Gaussian. Increasing the number of Gaussians per state is normally achieved using a simple greedy algorithm. To add an additional Gaussian to the model for a state, the Gaussian with the largest weight is selected and split into two new Gaussians, each with half the weight of the original. These are separated by perturbing the means by some proportion of the standard deviation. The parameters are then re-estimated by using the procedure above. This procedure is known as “mixing up”.

As the number of Gaussians is increased, some components may have very low training data counts. In this situation, even when covariances are constrained to be diagonal, the component variances computed from the training samples may be very small, leading to over-fitting to the training data. A standard remedy to this (Young *et al.*, 2006) is to use variance flooring: typically each diagonal variance element is floored at some fixed proportion (say 10%) of the mean within-state variance for that dimension, which may be computed globally. As an alternative, components with very low training data counts may be pruned. Further problems occur with covariance estimation in limited-data situations when covariances are not constrained to be diagonal: these are the subject of discussion in chapters 3, 4 and 5.

It is worth noting that the greedy algorithms for Gaussian mixture fitting do not guarantee a globally optimal solution for any fixed number of Gaussians. In addition, choosing the number of Gaussians is a hard problem: simply using a maximum likelihood approach would lead to a number of Gaussians equal to the number of points of training data, with one Gaussian centred on each point, which would generalise very poorly to unseen data. In practice, the number is often set by trial and error using development data, or simply fixed at some standard number, for example, 12 or 16 per state. We discuss this further in Chapter 3.

## 2.4 Acoustic modelling refinements

Here we discuss a selected set of acoustic modelling refinements, being the ones that we employ in our systems.

### 2.4.1 Dimensionality reduction

Having earlier discussed front-end feature processing, we now describe model-dependent feature transforms. Linear discriminant analysis (LDA) is a method for choosing a linear projection of the feature vector $\mathbb{R}^p \rightarrow \mathbb{R}^d$ $(d \leq p)$. LDA is an explicitly discriminative method: we seek a transform that maximises the ratio of the between-class variance to the within-class variance. In its classical formulation (see Duda & Hart, 1973), however, the assumption is made that all within-class covariances are equal.

Kumar & Andreou (1998) introduced heteroscedastic (linear) discriminant analysis (commonly referred to as HLDA), which removes the equal variance assumption, and derived a method for estimating the transform using the standard EM algorithm for parameter updates. We write the transformation as

$$o'_t = Ao_t \tag{2.38}$$

where $A$ is a $p$-dimensional square matrix, with the final $p-d$ rows corresponding to 'nuisance' dimensions which are removed from the final transformed feature vector: we denote this split by

$$A = \begin{pmatrix} A_d \\ A_{p-d} \end{pmatrix} \tag{2.39}$$

Since the nuisance dimensions are assumed to contain no class discrimination information, we model them with a $p-d$ dimensional global mean $\mu_{(g)}$ and covariance $\Sigma_{(g)}$. The remaining dimensions are modelled with Gaussian-specific $d$-dimensional parameters $\mu_m, \Sigma_m$. The parameters of the transformed vector $Ao_t$ are:

$$\mu_m^p = \begin{pmatrix} \mu_m \\ \mu_{(g)} \end{pmatrix}, \Sigma_m^p = \begin{pmatrix} \Sigma_m & 0 \\ 0 & \Sigma_{(g)} \end{pmatrix} \tag{2.40}$$

From (2.37), we change variables to obtain the log likelihood of the transformed vectors:

$$G(\theta, \theta^0) = -\frac{1}{2} \sum_m \sum_t \gamma_m(t) [\log |\Sigma_m^p| + (Ao_t - \mu_m^p)^T {\Sigma_m^p}^{-1} (Ao_t - \mu_m^p) - \log |A|^2] \tag{2.41}$$

(The final term is the Jacobian of the transformation). Holding $A$ fixed and maximising this expression with respect to $\mu^p$ and $\Sigma^p$, we obtain

$$\mu_m = A_d \bar{x}_m \tag{2.42}$$

$$\mu_{(g)} = A_{p-d} \bar{x} \tag{2.43}$$

$$\Sigma_m = A_d W_m {A_d}^T \tag{2.44}$$

$$\Sigma_{(g)} = A_{p-d} T {A_{p-d}}^T \tag{2.45}$$

where

$$\bar{x}_m = \frac{\sum_t \gamma_m(t) o_t}{\sum_t \gamma_m(t)} \qquad W_m = \frac{\sum_t \gamma_m(t)(o_t - \bar{x}_m)(o_t - \bar{x}_m)^T}{\sum_t \gamma_m(t)} \tag{2.46}$$

$$\bar{x} = \frac{\sum_m \sum_t \gamma_m(t) o_t}{\sum_m \sum_t \gamma_m(t)} \qquad T = \frac{\sum_m \sum_t \gamma_m(t)(o_t - \bar{x})(o_t - \bar{x})^T}{\sum_m \sum_t \gamma_m(t)} \tag{2.47}$$

are the within-class 2.46 and global 2.47 means and variances, respectively. Substituting these into (2.41) and ignoring constant terms, we obtain

$$G(\theta, \theta^0) = -\frac{1}{2} \sum_m \beta_m [\log |A_d W_m {A_d}^T| + \log |A_{p-d} T {A_{p-d}}^T| - \log |A|^2] \tag{2.48}$$

where $\beta_m = \sum_t \gamma_m(t)$. The transform $A$ is found to maximise this function. A method for this optimisation was developed by Gales (1997, 1999) in the context of semi-tied covariance matrices. We discuss this in Chapter 3.

### 2.4.2 Speaker adaptation

As mentioned in Chapter 1, much of the variability in the acoustic realisation of phonemes is due to speaker-specific variation, which cannot all be removed by the front-end normalisation described in Section 2.2.1. It would be desirable to train speaker-specific model parameters. This presents two difficulties:

- for most applications, it is likely that the speakers encountered in test data do not appear in the training set;
- the data available for an individual speaker may not have good phonetic coverage.

The limited data necessitates an adaptive approach: rather than training a new parameter set for each new speaker, we modify a speaker-independent set of parameters to be more suitable for that speaker. In some situations it may be possible to perform supervised adaptation by recording prescribed utterances from the target speaker; however, this is often not possible, and adaptation must be performed using the test data in an unsupervised manner: the new data is first transcribed using the initial speaker-independent model set and adaptation is performed using this reference transcription. This process can be iterated.

In Maximum A Posteriori (MAP) adaptation (Gauvain & Lee, 1994), the speaker-independent model set is used as a prior for the speaker-dependent model. By choosing the density of the prior appropriately, the speaker-dependent parameters may be obtained by a linear combination of the independent parameters and the speaker data. For example, for the means of a speaker $s$,

$$\mu_m^{(s)} = \frac{\tau\mu_m + \sum_t \gamma_m(t) o_t}{\tau + \sum_t \gamma_m(t)} \tag{2.49}$$

where the sum is over the adaptation data for speaker $s$, and the weights $\gamma_m$ have been obtained from the transcription obtained with the original parameter set. The MAP approach has the attractive property that as the amount of adaptation data reduces to zero, the adapted parameters tend to the original speaker-independent parameters. However, this does not solve the problem that the adaptation data is required to have good phonetic coverage for effective adaptation.

An alternative approach (Gales & Woodland, 1996; Leggetter & Woodland, 1995) is to use a set of linear transforms to adapt the speaker-independent parameters:

$$\mu_m^{(s)} = A^{(s)}\mu_m + b^{(s)} \tag{2.50}$$

$$\Sigma_m^s = B^{(s)}\Sigma_m B^{(s)^T} \tag{2.51}$$

where $A^{(s)}$ and $B^{(s)}$ are speaker-specific linear transforms and $b^{(s)}$ is a speaker-specific bias vector. The transforms can be trained using maximum likelihood – the technique is known as Maximum Likelihood Linear Regression (MLLR). The technique has the advantage that the linear transforms may be readily shared across Gaussians, so that all Gaussians may be transformed, regardless of the amount of adaptation data. Typically only a small number of transforms are used, with Gaussians clustered using a regression class tree. Constraining the mean and variance transforms to be equal, $A^{(s)} = B^{(s)}$ (Digalakis *et al.*, 1995; Gales, 1998) is known as Constrained MLLR (CMLLR). This has the advantage that the parameter transformation can be formulated as a transformation of the feature vector:

$$o_t' = A^{(s)}o_t + b^{(s)} \tag{2.52}$$

The translation term can be incorporated into a single transform by extending the feature vector and transform:

$$\zeta_t = \begin{pmatrix} 1 \\ o_t \end{pmatrix}, R^{(s)} = \begin{pmatrix} b & A^{(s)} \end{pmatrix} \tag{2.53}$$

so that (2.52) becomes

$$o_t' = R^{(s)}\zeta_t \tag{2.54}$$

To find the maximum likelihood transform $R$, we use a similar procedure to HLDA. The equivalent of Equation 2.41 is

$$G(\theta, \theta^0) = -\frac{1}{2}\sum_m \sum_t \gamma_m(t)[\log|\Sigma_m| + (R\zeta_t - \mu_m)^T\Sigma_m^{-1}(R\zeta_t - \mu_m) - \log|A|^2] \tag{2.55}$$

where the sum is over all adaptation data for speaker $s$. This gives an objective function

$$-\frac{1}{2}\sum_{m}\beta_m(\operatorname{tr}\Sigma_m^{-1}RS_mR^T - \log|A|^2) \tag{2.56}$$

where

$$S_m = \frac{\sum_t \gamma_m(t)(\zeta_t - \bar{\zeta}_m)(\zeta_t - \bar{\zeta}_m)^T}{\sum_t \gamma_m(t)},\ \bar{\zeta}_m = \frac{\sum_t \gamma_m(t)\zeta_t}{\sum_t \gamma_m(t)} \tag{2.57}$$

The optimisation of this function for the case when $\Sigma_m$ is diagonal is described in Gales (1998). The case when full covariance models are used has been considered by (Povey & Saon, 2006).

### 2.4.3 Speaker adaptive training

Speaker adaptive training (SAT) is a technique for explicitly allowing for inter-speaker variation during model training. The aim is to create a single model set that specifically does not model inter-speaker variation, allowing non-speaker sources of variation to be modelled more effectively. This speaker independent model set *must* be adapted to each test speaker in order to perform well.

SAT can be performed using an iterative procedure. An initial model set is trained and then adaptation transforms are trained for each training speaker. These transforms are used to “normalise” the data from each training speaker for a second iteration of model training. The procedure was suggested by Anastasakos *et al.* (1996). Gales (1998) derived efficient formulae for using CMLLR transforms for SAT.

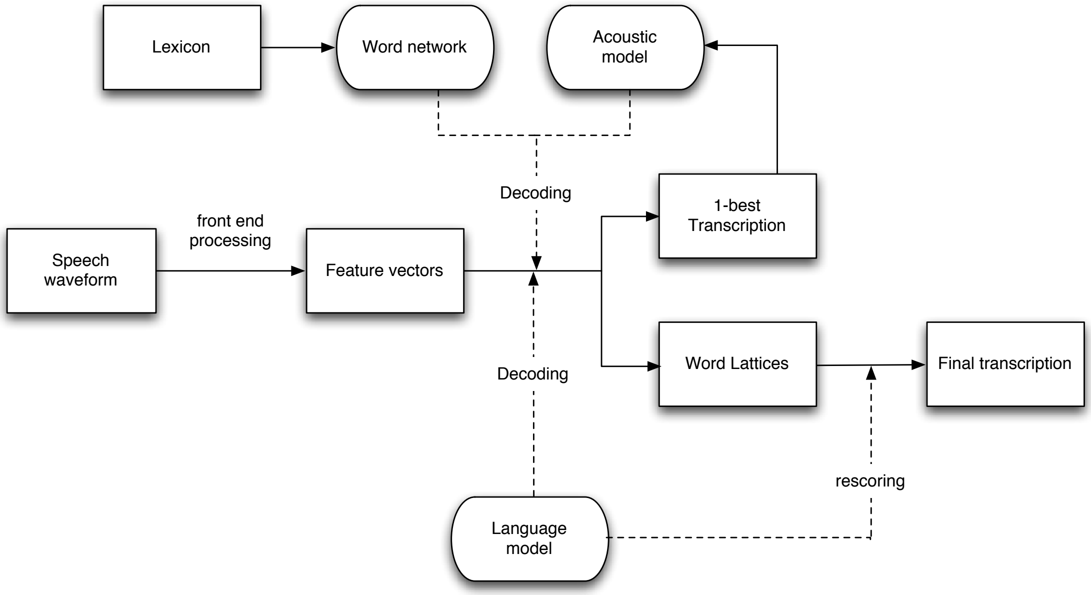

Figure 2.6: An illustration of an ASR system, incorporating several of the methods introduced in this chapter.

# Chapter 3

# Covariance modelling

In this chapter, we will explain the need for covariance modelling in an CD-HMM ASR system, and discuss the issues that arise during training and decoding that mandate the choice of model, and the method used to obtain its parameters. We review a range of methods used to resolve these issues.

## 3.1 Issues in covariance modelling

In a CD-HMM system, the need for covariance modelling arises (as we have seen in Chapter 2) in the computation of the Gaussian probabilities during decoding. For an observation vector $o_t$, the probability density of a Gaussian $m$ is given by

$$f_m(o_t) = f(o_t\ ;\ \mu_m, \Sigma_m) = (2\pi)^{-d/2}|\Sigma_m|^{-1/2}\exp\{-\frac{1}{2}(o_t - \mu_m)^T\Sigma_m^{-1}(o_t - \mu_m)\} \tag{3.1}$$

or using log probabilities and ignoring constants:

$$\log f_m(o_t) = -\frac{1}{2}\log|\Sigma_m| - \frac{1}{2}(o_t - \mu_m)^T\Sigma_m^{-1}(o_t - \mu_m) \tag{3.2}$$

Covariance modelling is concerned with obtaining a suitable $\Sigma_m$ for this computation, given some labelled training data. We use $U_m$ to denote the covariance matrix for a Gaussian $m$ obtained from the training data, using our model. In contrast to $\Sigma$, which we usually take to be some fixed but unknown parameter that describes the true distribution of the data, we view $U_m$ as a function of the available training data, that may have its form restricted in some way by limitations we impose on the model structure.

Suppose that $\Sigma_m^*$ is the (unknown) "optimal" covariance matrix to use (we will define what exactly we mean by "optimal" later). Then when selecting a covariance model, we broadly desire the following:

1. The model should enable an accurate value of $f_m(o_t)$ (from Equation 3.1) to be computed: in other words, $f_m(o_t\,; \mu_m, U_m)$ should be close to $f_m(o_t\,; \mu_m, \Sigma_m^*)$.

2. It should be possible to learn the model parameters, $U_m$, from a (possibly limited) quantity of training data.

3. Practically speaking, it should be computationally feasible to obtain the $U_m$, to store them in memory, and to compute $f_m(o_t\,; \mu_m, U_m)$.

In what follows, we sometimes suppress the dependence on $m$ for clarity.

Choosing a model requires a trade-off between the above properties. Assuming that $\Sigma_m^*$ has the maximum $\frac{1}{2}d(d+1)$ free parameters, we would expect to be able to compute $f_m(o_t)$ most accurately when $U_m$ has a maximum number of free parameters also. However, this does, of course, maximise the cost of storing the parameters and computing the densities. Furthermore, as the number of parameters increases, it becomes more difficult to reliably estimate them from limited training data. Of crucial importance, too, is the conditioning of the matrix $U_m$. If the matrix is ill-conditioned – that is to say, the ratio between the largest and smallest eigenvalues is large – then numerical errors are amplified when inverting the matrix or computing its determinant. Generally speaking, the chance of $U_m$ being ill-conditioned increases as the number of free parameters increases relative to the size of the training data.

As a motivating example, consider an artificial three-way classification problem in two dimensions, with data from each of the three classes distributed according as a single multivariate Gaussian, as shown in Figure 3.1. (Each class has an equal prior probability). We wish to classify the data using single-Gaussian parametric models for each class. Estimating the full covariance parameters from the data using maximum likelihood yields decision boundaries as shown in Figure 3.2. 1.2% of samples are incorrectly classified.

Now suppose that we use simpler, diagonal covariance models. If we estimate the parameters of the diagonal models by the diagonal elements of the full covariance matrices, the error rate is increased to 5.3%; the reduction in modelling

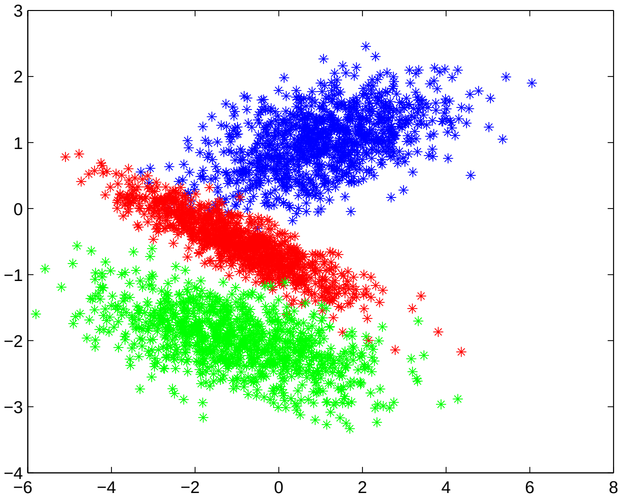

Figure 3.1: Samples from three classes, each drawn from a single Gaussian

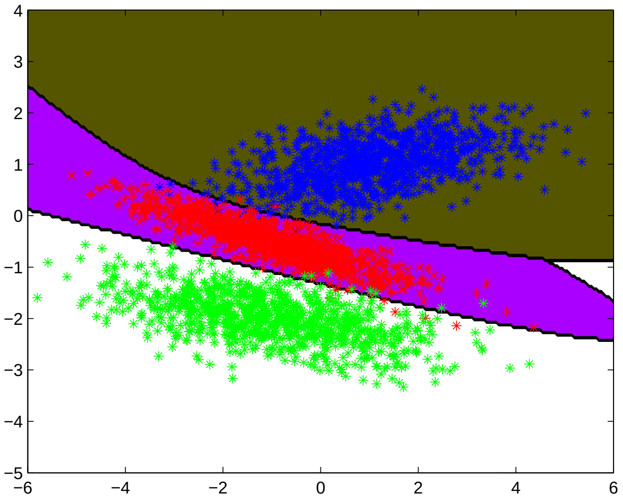

Figure 3.2: Samples from three classes, with decision boundaries obtained using maximum-likelihood full covariance Gaussian models

power limits the functional form of the new decision boundaries, which are shown in Figure 3.3.

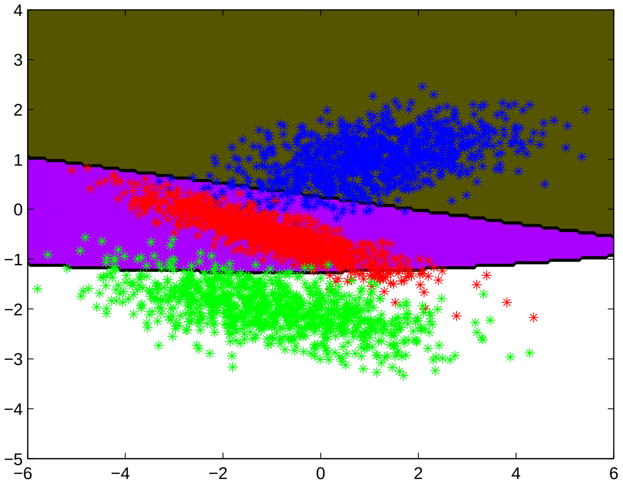

Figure 3.3: Samples from three classes, with decision boundaries obtained using maximum-likelihood diagonal covariance Gaussian models

This motivates the use of more complex covariance models than the diagonal models commonly used. However, a number of questions arise.

- Could the lack of sufficient complexity in covariance modelling be avoided by increasing model complexity elsewhere? For example, by increasing the number of Gaussians.
- The use of the complex models works well in this scenario, where there is sufficient data to estimate the parameters reliably – but what happens when data is limited?
- Are there learning methods that could be employed to allow the lower complexity models to achieve improved classification performance?

We will consider all these questions in the following chapters, and at times refer to the example presented here.

Although we will give regard in this chapter to the practical issues of parameter storage and density computation, modern high-performance computing facilities are such that these requirements do not pose any hard limits on the models that may be used – in experimental systems, at least. We have found that it is quite possible to train and decode with full-parameter matrices in large-vocabulary recognition systems. We therefore do not discuss these issues in detail. In the following sections, however, we consider further the problems of matrix conditioning and parameter estimation.

As a general point of notation, we use $o_t$, to refer to a $d$-dimensional acoustic feature vector, generated from a mixture of Gaussians. We use $x$, or $x_i$ to refer to an abstract feature vector generated from a single known Gaussian distribution – or occasionally to refer to an arbitrary observation of a continuous random variable.

### 3.1.1 GMM covariance estimation

We briefly describe the process of covariance estimation within an CD-HMM. We seek parameters for Gaussian $m$ to maximise the auxiliary function given in Equation 2.37 on page 25:

$$\begin{aligned} G(\theta,\theta^0) &= \sum_t \gamma_m(t)[-\frac{1}{2}\log|U_m| - \frac{1}{2}(o_t-\mu_m)^T U_m^{-1}(o_t-\mu_m)] && (3.3)\\ &= -\frac{1}{2}\beta_m \log|U_m| - \frac{1}{2}\operatorname{tr}[U_m^{-1}\sum_t \gamma_m(t)(o_t-\mu_m)(o_t-\mu_m)^T] && (3.4)\\ &= -\frac{\beta_m}{2}(\log|U_m| + \operatorname{tr} U_m^{-1}S_m) && (3.5) \end{aligned}$$

where $S_m$ is the sample covariance matrix, defined by

$$S_m = \frac{\sum_t \gamma_m(t)(o_t-\mu_m)(o_t-\mu_m)^T}{\beta_m} \tag{3.6}$$

We have used the fact that the trace of a scalar is that scalar, and that the trace operator is invariant to cyclic permutations.

For the purposes of estimating the covariance matrix by maximising this function, all that matters is that for each Gaussian $m$, each observation $o_t$ has been assigned a weighting $\gamma_m(t)$. If the motivation is to maximise the log likelihood of the observations using the EM algorithm, then $\gamma_m(t)$ is the posterior probability of $o_t$ having been drawn from the Gaussian $m$, given the whole observation sequence, $O$, and the previous parameter set. However, the analysis applies equally when the weights have been chosen in some other way – we discuss this in more detail below.

Setting $P_m = U_m^{-1}$, we can write (3.5) in precision matrix form:

$$G(\theta, \theta^0) = \frac{\beta_m}{2}(\log|P_m| - \operatorname{tr} P_m S_m) \tag{3.7}$$

Assuming a full covariance matrix, we maximise (3.5) with respect to $U_m$ by differentiating and setting the result equal to zero:

$$0 = U_m^{-1} - U_m^{-1} S_m U_m^{-1} \tag{3.8}$$

$$\Rightarrow U_m = S_m \tag{3.9}$$

Note that in this formulation, the statistic $S_m$ is dependent on $\mu_m$, which is typically an unknown parameter. However, it follows from differentiating (3.5) with respect to $\mu_m$, that it can be maximised by setting

$$\mu_m = \sum_t \frac{\gamma_m(t) o_t}{\beta_m} \tag{3.10}$$

independently of the eventual choice of $U_m$ – or alternatively, $\mu_m$ may be set to some previously-determined value.

It is not necessary that the weights $\gamma_m(t)$ are the posterior probabilities (although of course it is necessary to ensure that the weights used have the effect that the auxiliary results in an increase in the desired objective function at the end of the iteration). As an example, when using a discriminative objective function such as MMI (Bahl *et al.*, 1986) we would set

$$\gamma_m(t) = \gamma_m^n(t) - \gamma_m^d(t) \tag{3.11}$$

where $\gamma_m^n(t)$ and $\gamma_m^d(t)$ are the posterior probabilities given the correct models and all possible models, respectively. This is described in more detail in Chapter 7.

However, one problem arises here: in the case where some of the $\gamma_m(t)$ are negative (which is of course possible from equation 3.11), $S_m$ is no longer guaranteed to be positive semidefinite (see below). However, it is possible to add a smoothing term to (3.3) to ensure a positive definite matrix (Normandin & Morgera, 1991); again, we discuss this further in Chapter 7.

### 3.1.2 Matrix conditioning

A covariance matrix $\Sigma$ is not simply a collection of independent scalar parameters: it has a meaning in $d$-dimensional feature space. For the Gaussian distribution, the curve $x^T\Sigma^{-1}x = C^2$ (where $C$ is a constant) defines an ellipse in feature space. The principal axes of the ellipse are given by the eigenvectors of $\Sigma$, and the variance in the dimension given by each axis is given by the corresponding eigenvalue. An example is shown in Figure 3.4. This shows 500 samples from a 2-dimensional Gaussian distribution. The ellipses in black enclose the regions within one and two standard deviations of the mean. (Corresponding to $C = \{1, 2\}$).

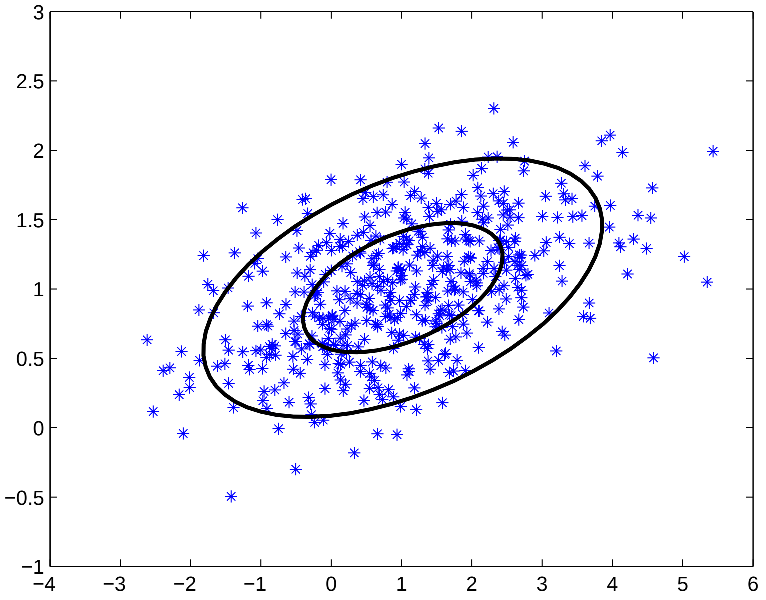

Figure 3.4: 500 samples from a 2-dimensional Gaussian distribution, with ellipses enclosing regions within 1 and 2 standard deviations of the mean.

Considering this interpretation, it is important to ensure that a covariance matrix $U$ is "well-behaved", in terms of the ellipse that it defines. We first define some terms.

A symmetric matrix $A$ is *positive definite* if $x^TAx > 0$ for all $x \in \mathbb{R}^d$; or that all eigenvalues $\lambda_i$ of $A$ are strictly positive, $\lambda_i > 0$. $A$ is *positive semidefinite* if

$v^T Av \geq 0$ for all $v \in \mathbb{R}^d$; or that all $\lambda_i \geq 0$. We denote the set of symmetric $d$-dimensional matrices by $\mathbb{S}^d$, the set of positive semidefinite matrices by $\mathbb{S}^d_+$, and the set of positive definite matrices by $\mathbb{S}^d_{++}$. Note that if $\Sigma$ is positive definite, then it is invertible, and its inverse is also positive definite, with eigenvalues given by $\lambda_i^{-1}$.

The property of positive-definiteness can be used to define a partial ordering on the set of symmetric matrices. We write

$$\begin{aligned} A \preceq B \quad &\text{if} \quad B - A \in \mathbb{S}^k_+ \\ A \prec B \quad &\text{if} \quad B - A \in \mathbb{S}^k_{++} \end{aligned}$$

If $A \preceq B$ then we can also write $B \succeq A$, and so on. The notation $A \succ 0$ is often used to denote that $A$ is positive definite. Any covariance matrix $\Sigma$ is clearly symmetric. The physical interpretation given earlier in this section implies that it is also positive definite.

We define the *condition number* of a symmetric, positive semidefinite matrix $A$ to be the ratio of the largest and smallest eigenvalues:

$$\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \tag{3.12}$$

with $\kappa(A) = \infty$ when $\lambda_{min}(A) = 0$. (For a wider class of matrices, the condition number may be expressed as a ratio of maximum and minimum singular values, but this is not necessary here). The condition number becomes important when a matrix is inverted, as it is in the computation of the Gaussian probability density. From a theoretical perspective, suppose a vector $x$ has been obtained with error $e$. Then we consider the error in $x^T A^{-1} x$ relative to the error in $x$:

$$\begin{aligned} &\frac{|(x+e)^T A^{-1}(x+e) - x^T A^{-1} x| / |x^T A^{-1} x|}{\|(x+e) - x\| / \|x\|} && (3.13) \\ &\approx \frac{2|x^T A^{-1} e|}{\|e\|} . \frac{\|x\|}{|x^T A^{-1} x|} && (3.14) \\ &= 2 \frac{|x^T A^{-1} e|}{\|x\|} . \frac{\|x\|}{|x^T A^{-1} x|} && (3.15) \\ &\leq 2.\lambda_{min}^{-1} \|x\| . \frac{1}{\lambda_{max}^{-1} \|x\|} = 2 \frac{\lambda_{max}}{\lambda_{min}} && (3.16) \end{aligned}$$

so the error in computing $f(o_t)$ is directly related to the condition number of the matrix $U$ used in the computation. Practically speaking, standard matrix inversion algorithms fail to operate well when $U$ is ill-conditioned. Of course, if $U$ is singular (when one or more of the eigenvalues are zero) then $f(o_t)$ cannot even be computed. We say that a matrix is *well-conditioned* when the condition number is small.

Suppose that $U$ is set to the sample covariance matrix $S$, as defined in Equation 3.6. Note that $S$ is positive semidefinite. However, $S$ will have some eigenvalues equal to zero if the number, $n$, of linearly independent sample observations $o_t$ for which $\gamma(t)$ is non-zero, is less than $d$. For Gaussians used in ASR systems, $d$ is typically 39, and could even be 52 – so this is a practical consideration in systems with relatively large numbers of Gaussians and small amounts of data.

Moreover, consider the case when $S$ is an unbiased estimator for a true matrix $\Sigma$. It can be shown that on average, the eigenvalues of $S$ are more dispersed about their mean than the eigenvalues of $\Sigma$, and so we expect the sample covariance matrix to be less well-conditioned than the true covariance matrix. We derive this result in Appendix A.2; it follows from the fact that the eigenvalues are the most dispersed diagonal elements of $R^TSR$ for any rotation $R$. This suggests that choosing $U = S$ for our covariance model would be a bad choice if a well-conditioned $U$ is desired.

We end this section with a simple illustration. The most popular covariance model for ASR systems is to set $U$ to be the diagonal elements of the sample covariance matrix, corresponding to the variance in each dimension of feature space. We denote this by $D$. The diagonal covariance matrix will:

1. Almost always be non-singular
2. Always be at least as well-conditioned as $S$, and almost always better conditioned.

(1) holds when all the diagonal elements are non-zero. A diagonal element will only be zero if the samples are identical in that dimension, something we would expect to occur only with very small sample sizes. (2) is true because the eigenvalues of $D$ are just the diagonal elements themselves, so $\mathcal{D}(D)$ is given by the dispersion of the diagonal elements of $S$. Earlier, however, we showed that the diagonal elements of $S$ are less dispersed than its eigenvalues, with equality occurring only when $S$ itself is diagonal (which would occur when all features were perfectly uncorrelated).

Figure 3.5 gives an illustration. With a very small number of samples (but still with $n > d$), the diagonal covariance matrix, shown with red dashes, has a much smaller variation in length between its two principal axes than the full sample covariance matrix, shown in solid black.

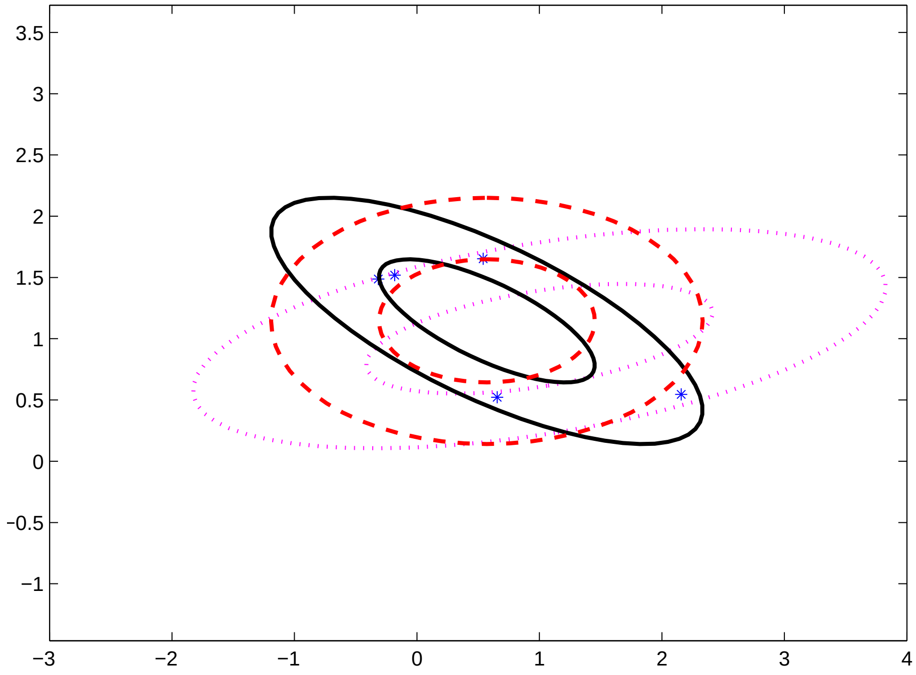

Figure 3.5: 5 samples from a 2-dimensional Gaussian distribution, with ellipses enclosing regions within 1 and 2 standard deviations of the mean for: true covariance matrix (dots); diagonal sample covariance matrix (dashes) and full sample covariance matrix (solid)

### 3.1.3 Generalisation

Generalisation, in the context of statistical learning, refers to the ability of a model whose parameters are learnt from limited training data to work appropriately when applied to unseen test data. How can this notion be expressed more formally? Suppose that both our training and test data are selected randomly from a single pool of all possible data that could exist in the given domain. We assume that this data has probability density $p(x, y)$ ($x$ being the observed features, $y$ the labels). In training, we want each matrix $U$ computed from this random training data to be close to the matrix that would have been used for the random test data. However, the test data is not known when the parameters are learnt. The best we could do, therefore, is to find the $U$ that is *most likely* to be appropriate, according to the distribution of test data. This could be achieved with perfect knowledge of $p(x, y)$, which we would attain as the amount of training data approached infinity. Let $L_U(x, y)$ be a measure of the loss incurred by using a matrix $U$ when the data presented is $(x, y)$. The risk, $R(U)$ is defined as the expected loss under the true distribution of data:

$$R(U) = \mathbb{E}_{p(x,y)} L_U(x, y) \tag{3.17}$$

The optimal matrix $U^*$ would be chosen to minimise this risk, within the constraints of the model:

$$U^* = \arg\min_U R(U) \tag{3.18}$$

In practice, we do not know the distribution $p(x, y)$, we can instead attempt to minimise an estimate of the risk, called the *empirical risk* on the training set Vapnik (1995). For a training set of size $n$, this is given by

$$R_{\text{emp}} = \frac{1}{n} \sum_r^n L_U(x_r, y_r) \tag{3.19}$$

We can obtain $U(n)$, the (random) matrix obtained from $n$ items of training data by

$$U_n = \arg\min_U R_{\text{emp}}(U) \tag{3.20}$$

A model is said to generalise well if we expect that, for any chosen training data, the learnt parameters $U_n$ will give a performance close to that of $U^*$ – if the empirical risk is close to the actual risk. In practice, we do not know the distribution $p(x, y)$ or the parameters $U^*$, so how is this notion helpful? Though there are theoretical bounds on the gap between the two functions (Vapnik, 1995), in broad terms, we know that the generalisation ability of a trained model is higher if:

- the number of samples is large;
- the number of parameters is small.

A model is often said to be *over-fitted* to training data if the number of parameters is too large relative to the amount of training samples, leading to worse performance on test data – the model generalises poorly.

Full covariance models have $\frac{1}{2}d(d+1)$ parameters per Gaussian, compared to just $d$ mean parameters per Gaussian, and are thus much more susceptible to over-fitting. So how can we ensure that a covariance model has good generalisation for a fixed amount of training data? Techniques can be placed in the following categories:

- restrict the dimensionality of the model by fixing the values of some parameters;
- share the parameters over multiple Gaussians, thus increasing the amount of data available to train each parameter;
- modify the learning method to control the number of parameters automatically – for example, by including penalty terms in the objective function – or to explicitly improve generalisation.

Some techniques may fall into more than one category.

### 3.1.4 Types of covariance model

In the following sections we describe a variety of covariance models. These can be broadly placed in two categories, according to whether they explicitly model the covariance matrix, or its inverse, the (true) precision matrix, $\Omega_m = \Sigma_m^{-1}$. The distinction is motivated by the observations that, when modelling the covariance matrix using $U_m$:

1. $U_m$ must typically be learnt from sample data via the statistic $S_m$
2. $U_m$ is required for the computation of $f_m(o_t)$ during decoding. From equation 3.2, we have

$$\log f_m(o_t\,; \mu_m, U_m) = -\frac{1}{2}\log|U_m| - \frac{1}{2}(o_t - \mu_m)^T U_m^{-1}(o_t - \mu_m) \qquad (3.21)$$

$$= \frac{1}{2}\log|P_m| - \frac{1}{2}(o_t - \mu_m)^T P_m(o_t - \mu_m) \qquad (3.22)$$

   where $P_m = U_m^{-1}$.

It can be seen that we have a choice between modelling $U_m$ directly – in which case, we would expect each matrix to be readily obtained from sample data – or, instead, modelling its inverse, $P_m$. In the latter case, we need never find an expression for $U_m$ itself, and the computation of $f_m(o_t)$ would be simplified, at the cost of greater complexity in estimating $P_m$ from data. Since $P_m$ approximates the precision matrix $\Omega_m$, $P_m$ is termed a *precision matrix model.*

## 3.2 Simple approximations

### 3.2.1 Diagonal matrices

Most GMM systems for ASR model the covariance matrices as diagonal matrices, implicitly assuming that the features are uncorrelated, given $m$. Under this restriction, the auxiliary function (3.3) is maximised by setting

$$U_m = \operatorname{diag}(S_m) \tag{3.23}$$

We denote this diagonal model by $D_m$. This diagonal approximation has both practical and theoretical advantages:

- $D_m$ has low variance compared to the full matrix $S_m$, so has good generalisation.
- As discussed in Section 3.1.2, $D_m$ is always better conditioned than $S_m$, and on average, is better conditioned than $\Sigma_m$.
- Storing $D_m$ requires storing only $d$ parameters per Gaussian.
- It is trivial to invert $D_m$, and the inverted version has the same number of parameters. It is therefore easy to use during decoding.

### 3.2.2 Block-diagonal matrices

Use of block-diagonal covariance matrices is a compromise between the full covariance and diagonal covariance cases. Here, the feature vector is partitioned into sets. Features within each set are assumed fully correlated; features in different sets are assumed independent. Re-ordering the features so that the elements within a set are adjacent, the covariance matrix has block diagonal form:

$$U_m = \begin{pmatrix} B_1^{(m)} & \mathbf{0}_{(d_1 \times d_2)} & \cdots & \mathbf{0}_{(d_1 \times d_n)} \\ \mathbf{0}_{(d_2 \times d_1)} & B_2^{(m)} & \cdots & \mathbf{0}_{(d_2 \times d_n)} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}_{(d_n \times d_1)} & \mathbf{0}_{(d_n \times d_2)} & \cdots & B_n^{(m)} \end{pmatrix} \tag{3.24}$$

where $d_i$ is the cardinality of the $i^{th}$ set, $\sum_i d_i = d$, and $B_i$ is the covariance matrix for the $i^{th}$ set. The standard case is that this correlation structure is the same for all Gaussians.

A block-diagonal matrix may again be simply estimated from the sample covariance matrix, constraining the relevant entries to zero. The use of the block-diagonal matrices gives more modelling power than the diagonal covariance case, but maintains advantages over using full covariance matrices:

- The block-diagonal matrix has fewer parameters than the full covariance matrix, so the estimator has lower variance.
- The block-diagonal matrix is better conditioned than the full matrix. In particular, the matrix is invertible if there are at least $\max_i d_i$ samples.
- Inverting the matrix can be achieved by inverting each block independently, and is more efficient than inverting the full matrix. Moreover, the inverse matrix has the same block diagonal structure.

Block-diagonal schemes used in ASR tend to use prior knowledge to specify the covariance structure. Typically, the feature vector is partitioned into sets corresponding to the static features, and first and second differentials. This does, however, require the generally false assumption that features are not correlated with their respective differentials.

## 3.3 Parameter tying and basis decomposition

### 3.3.1 Covariance tying

A simple method for reducing the number of covariance parameters in the system is to tie all covariance parameters between Gaussians in a specified class. Suppose we have a class $r$, and a set $M(r)$ of Gaussians that belong to it. Then the tied covariance may be estimated by

$$U^r = \frac{\sum_{m\in M(r)}\sum_t \gamma_m(t)(o_t-\mu_m)(o_t-\mu_m)^T}{\sum_{m\in M(r)}\sum_t \gamma_m(t)} \tag{3.25}$$

This increases the amount of data available to estimate each matrix, so makes the estimates more robust. However, this is at the cost of reduced inter-Gaussian discrimination. It is preferable to preserve Gaussian-specific parameters where possible: a diagonal covariance system of Gaussians is to be preferred over a system with more covariance parameters, but with those parameters shared over multiple Gaussians. Semi-tied covariance matrices, explained below, is an improved parameter tying scheme where the number of covariance parameters is increased from a diagonal system, without the number of Gaussian-specific parameters being reduced.

### 3.3.2 Semi-tied covariance matrices

(Gales, 1999) proposed a scheme for decomposing a covariance matrix into a Gaussian-specific diagonal matrix and a class-specific transformation. This scheme is known as Semi-tied covariance matrices (STC). Again denoting the tied class by $r$, the covariance matrix for Gaussian $m$ is given by

$$U_m = H^{(r)}\Lambda_m H^{(r)T} \tag{3.26}$$

where $\Lambda_m$ is a diagonal matrix with diagonal elements $(\sigma_1^{(m)2},, \sigma_2^{(m)2}, \ldots, \sigma_d^{(m)2})$ Writing $A^{(r)} = H^{(r)-1}$, the auxiliary function (see Equation 3.3) is given by

$$G(\theta,\theta^0) = -\frac{1}{2}\sum_{m\in M(r)}\sum_t \gamma_m(t)[\log|\Lambda_m| - \log|A^{(r)}|^2 + (o_t-\mu_m)^T A^{(r)T}\Lambda_m^{-1}A^{(r)}(o_t-\mu_m)] \tag{3.27}$$

This can be compared to the auxiliary functions in equations 2.41 and 2.55, for HLDA and CMLLR respectively. It is clear that $A^{(r)}$ can be viewed as a feature-space linear transform – the technique is also known as a maximum-likelihood linear transform (MLLT) model. The equation can be re-written as

$$G(\theta,\theta^0) = -\frac{1}{2}\sum_{m\in M(r)} \beta_m[\log|\Lambda_m| - \log|A^{(r)}|^2 + \mathrm{tr}\,\Lambda_m^{-1}A^{(r)}S_m A^{(r)T}] \tag{3.28}$$

The optimal diagonal variances $\Lambda_m$ and transform $A^{(r)}$ are dependent on each other, but it is not possible to optimise them jointly. Instead, the parameters are iteratively updated. Using an initial transform (usually just an identity transform), we obtain

$$\Lambda^m = \mathrm{diag}(A^{(r)} S_m A^{(r)T}) \tag{3.29}$$

To find the optimal transform, it is helpful to re-express the transformed precision matrix:

$$P_m = A^{(r)T} \Lambda_m^{-1} A^{(r)} = \sum_i^d \frac{1}{\sigma_i^{(m)2}} a_i^T a_i \tag{3.30}$$

where $a_i$ denotes the $i^{th}$ row of $A^{(r)}$. (The superscript $r$ is dropped for notational clarity). Then (3.27) can be rewritten as

$$G(\theta, \theta^0) = -\frac{1}{2} \sum_{m \in M(r)} \beta_m \left[ \log|\Lambda_m| - \log|A^{(r)}|^2 + \mathrm{tr} \sum_i a_i^T a_i \frac{1}{\hat{\sigma}_i^{(m)2}} S_m \right] \tag{3.31}$$

We therefore maximise

$$\beta \log|A^{(r)}|^2 - \mathrm{tr} \sum_i a_i^T a_i K^{(ri)} \tag{3.32}$$

where

$$K^{(ri)} = \sum_{m \in M(r)} \frac{\beta_m}{\hat{\sigma}_i^{(m)2}} S_m, \quad \beta = \sum_{m \in M(r)} \beta_m \tag{3.33}$$

The expression is iteratively maximised with respect to each $a_i$ with the other rows of the transformation held constant. (See Gales, 1999). The maximisation requires the inverse of $K^{(ri)}$ to be computed. When data is limited, these matrices may be poorly-conditioned (although this is unlikely if the set $M(r)$ is chosen to be large enough). Gales (1999) constrained the matrices to be block-diagonal, reducing the likelihood of them being poorly-conditioned.

The STC scheme represents the precision matrix in a compact form as a sum of $d$ basis elements. This allows efficient computation of the likelihood for the Gaussian; the Jacobian term $\log|A^{(r)}|^2$ need only be evaluated once for each semi-tied class.

### 3.3.3 Precision matrix subspace methods

Precision matrix subspace models represent a general class of precision matrix models. Each precision matrix is decomposed as the linear combination of a global set of $k$ basis elements, given by symmetric matrices $W_i$, and $k$ Gaussian-specific coefficients, $\lambda_i^{(m)}$

$$P_m = \sum_{i}^{k} \lambda_i^{(m)} W_i \tag{3.34}$$

Provided that $k < \frac{1}{2}d(d+1)$, this represents a reduced dimensionality model compared to the use of a full covariance model for each state.

The STC scheme with a single semi-tied class can be viewed as precision matrix subspace model. Consider the STC precision matrix decomposition in Equation 3.30:

$$P_m = \sum_{i}^{d} \frac{1}{\sigma_i^{(m)2}} a_i^T a_i \tag{3.35}$$

Comparing to 3.34, we can see this is a subspace model with dimensionality $k = d$, with rank-1 positive definite basis matrices $W_i = a_i^T a_i$, and positive basis coefficients $\lambda_i = \frac{1}{\sigma_i^{(m)-2}}$.

Extended MLLT (Olsen & Gopinath, 2004) is a natural extension to this scheme where the number of basis matrices is increased from $d$, up to the maximum size of the space $\mathbb{S}_+^d$, $\frac{d}{2}(d-1)$. A single set of basis matrices across all Gaussians is assumed. The EMLLT scheme allows a smooth increase in the number of Gaussian-specific covariance parameters up to the full covariance case. Since the precision matrix is modelled directly, decoding with EMLLT models is efficient. However, no closed-form solution exists for updating the basis matrices when the dimension of the space is greater than $d$.

The precision-constrained GMM (PCGMM) and subspace for precision and mean (SPAM) schemes (Axelrod *et al.*, 2005) are a further generalisation of EMLLT, where the basis elements are arbitrary symmetric matrices (of any rank). In this case the precision matrices are not automatically guaranteed to be positive definite, and this must be explicitly ensured when the per-Gaussian coefficients are optimised. Axelrod *et al.* (2005) found these schemes to give improved performance over EMLLT on ASR tasks.

### 3.3.4 Factor analysis

Factor analysis reduces the dimensionality of the covariance matrix by modelling the observation $o_t$ as being generated by a lower-dimensional intermediate vector of "factors", $x_t$, via a linear transform, with the addition of a noise term:

$$o_t = Cx_t + v_t \tag{3.36}$$

The vector $x_t$ is unobserved. The simplest case is when there is just one Gaussian per state, and $x$ is modelled by a single Gaussian too. Without loss of generality, we can assume that $x_t$ has zero mean and unit variance, $x_t \sim \mathcal{N}(0, I)$, and $v_t \sim \mathcal{N}(\mu_j^{(o)}, \Lambda_j^{(o)})$. It is important that $\Lambda_j^{(o)}$ is a diagonal matrix.

$$p(o_t|x_t, q_t = j) = f(o_t; \mu_j^{(o)} + C_j x_t, \Lambda_j^{(o)}) \tag{3.37}$$

The covariance matrix for the Gaussian $j$ is then given by

$$\Sigma = C_j C_j^T + \Lambda_j^{(o)} \tag{3.38}$$

If $x$ has dimension $k$, then the covariance matrix has $d(k + 1)$ parameters, so we require $k < \frac{1}{2}(d - 1)$ for a reduction in parameter number. It is possible to find the parameters $C_j$ and $\Lambda_j^{(o)}$ maximising the likelihood of the data by finding the eigenvalues of the sample covariance matrix (Stoica & Jansson, 2009). $C_j$ is set to be the matrix of eigenvectors corresponding to the largest $k$ eigenvalues, scaled by the respective eigenvalues. However, this has some drawbacks: firstly, it requires good initial estimates of the eigenvalues; secondly, it is not invariant to arbitrary scaling of the feature dimensions.

The Factor-Analysed Covariances Invariant to Linear Transforms (FACILT) scheme (Gopinath *et al.*, 1998) was proposed as an extension to this scheme where a linear transformation is applied to the vector $v_t$; the authors derived an EM algorithm for estimating this transformation and the parameters $C_j$ and $\Lambda_j^{(o)}$ in an HMM setting, when they are shared independently between Gaussians or states.

As an alternative extension to the above formulation, Rosti & Gales (2004) proposed the factor-analysed HMM (FAHMM). Both the vector $x_t$ and the noise term $v_t$ are drawn from independent GMMs. The factors, and transform $C_j$ are shared between all Gaussians for the state. We set $\omega_t^{(x)}$ to be the random variable setting the Gaussian for $x_t$, indexed by $n$, and $\omega_t^{(o)}$ to be the equivalent for $v_t$, indexed by $m$.

$$p(x_t|q_t = j) = \sum_n^N c_{jn} f(x_t; \mu_{jn}^{(x)}, \Lambda_{jn}^{(x)}) \tag{3.39}$$

$$p(o_t|x_t, q_t = j) = \sum_m^M c_{jm} f(o_t; \mu_{jm}^{(o)} + C_j x_t, \Lambda_{jm}^{(o)}) \tag{3.40}$$

The total number of mean and covariance parameters is $2Nk + 2Md + dk$. The distribution of $o_t$, given $\omega_t^{(x)}$ and $\omega_t^{(o)}$, is another Gaussian:

$$p(o_t|q_t = j, \omega_t^{(x)} = n, \omega_t^{(o)} = m) = f(o_t; \mu_{jmn}, \Sigma_{jmn}) \tag{3.41}$$

where

$$\mu_{jmn} = C_j \mu_{jn}^{(x)} + \mu_{jm}^{(o)} \tag{3.42}$$

$$\Sigma_{jmn} = C_j \Lambda_{jn}^{(x)} C_j^T + \Lambda_{jm}^{(o)} \tag{3.43}$$

The covariance matrix for the joint system is given by

$$\text{var}\left[\begin{pmatrix} x_t \\ o_t \end{pmatrix}\right] = \begin{pmatrix} \Lambda_{jn}^{(x)} & \Lambda_{jn}^{(x)} C_j^T \\ C_j \Lambda_{jn}^{(x)} & \Sigma_{jmn} \end{pmatrix} \tag{3.44}$$

from which it follows that the conditional distribution of $x_t$ is

$$p(x_t|o_t, q_t = j, \omega_t^{(x)} = n, \omega_t^{(o)} = m) = f(x_t; \mu_{jmn}^{(x|o)}, \Sigma_{jmn}^{(x|o)}) \tag{3.45}$$

with

$$\mu_{jmn}^{(x|o)}(t) = \mu_{jn}^{(x)} + \Lambda_{jn}^{(x)} C_j^T \Sigma_{jmn}^{-1} (o_t - C_j \mu_{jn}^{(x)} - \mu_{jm}^{(o)}) \tag{3.46}$$

$$\Sigma_{jmn}^{(x|o)} = \Lambda_{jn}^{(x)} - \Lambda_{jn}^{(x)} C_j^T \Sigma_{jmn}^{-1} C_j \Lambda_{jn}^{(x)} \tag{3.47}$$

The parameters may be jointly optimised using the EM algorithm. The transform $C_j$ is optimised using a similar method to STC.

A problem with the FAHMM is that the inverse matrix $\Sigma_{jmn}^{-1}$, required for decoding, does not have a compact representation. If the inverses are pre-computed then decoding is as expensive, in terms of computation and memory, as when unconstrained full covariance models are used. However, the matrix may be inverted using the formula

$$\Sigma_{jmn}^{-1} = \Lambda_{jm}^{(o)-1} - \Lambda_{jm}^{(o)-1} C_j (C_j^T \Lambda_{jm}^{(o)-1} C_j + \Lambda_{jn}^{(x)-1})^{-1} C_j^T \Lambda_{jm}^{(o)-1} \tag{3.48}$$

which requires inverting a single $k$-dimensional matrix, $C_j^T \Lambda_{jm}^{(o)-1} C_j + \Lambda_{jn}^{(x)-1}$, rather than a full $d$-dimensional matrix. This allows a compromise between inverting the matrices as required during decoding, with higher computational cost, or pre-computing them, with higher memory cost. Evaluating the likelihood requires $\mathcal{O}(dk)$ computations.

## 3.4 Full covariance models

Full covariance models use the maximum $\frac{d}{2}(d+1)$ untied covariance parameters per Gaussian, giving the highest possible discriminative power. These models have the following properties:

- Parameter estimation is simple to achieve: the maximum likelihood estimator is just the sample covariance matrix $S_m$. There is no need for the complicated optimisation schemes required for EMLLT and SPAM models.
- The models are expensive in terms of the memory needed for parameter storage and the computational cost of decoding: both are $\mathcal{O}(d^2)$.
- Large amounts of training data are required for reliable full covariance estimation: otherwise, as we discussed in earlier sections, the matrices are often poorly-conditioned, and do not generalise well.

Despite the shortcomings above, full covariance systems have been successfully used for large vocabulary ASR, the most notable example being in the 2004 IBM system (Chen *et al.*, 2006; Soltau *et al.*, 2005), where the computational cost was reduced by aggressively pruning Gaussians during the full covariance likelihood computation.

In their comprehensive review of covariance modelling, Axelrod *et al.* (2005) conclude that full covariance models achieve the highest performance. In addition, recent advances in computing mean that the requirements imposed by full covariance models no longer impose hard constraints on the systems that can be built. We therefore consider full covariance modelling to be a promising direction for research, and in the following chapters of this thesis, focus mainly on resolving the final point above.

## 3.5 A note on choosing the number of mixture components

Choosing the number of Gaussians to use in the mixture models is a research question with implications for covariance modelling. There have been a number of theoretically-motivated methods proposed for choosing the optimal number of components, based, for example, on the Bayesian information criterion (Schwarz, 1978), or using discriminative growth functions (Liu & Gales, 2007). However, systems often optimise the number of components by measuring the performance on held-out data, or simply use a preset number – for example, 16 Gaussians per state.

Using multiple Gaussians can act as a proxy for increasing the number of covariance parameters by implicitly modelling feature correlations, and we would expect fewer covariance parameters to be required in systems with more Gaussians. However, it is not clear how this trade-off could easily be optimised, particularly when increasing the number of covariance parameters incrementally, for example by increasing the number of classes in STC or the number of basis matrices in the EMLLT and PCGMM schemes. In our phone-recognition experiments in Chapter 6 we briefly investigate this issue experimentally. In our large vocabulary system, however, we simply use the same number of Gaussians for the full covariance system as in the baseline diagonal system, which we assume has been previously optimised.

An important question is whether all the performance gains derived from increasing the number of covariance parameters could be achieved (perhaps even more cheaply) by increasing the number of Gaussians. This was investigated in the context of large-vocabulary ASR by Axelrod *et al.* (2005), who conclude that they cannot: they find that a full covariance system with 10,000 Gaussians outperforms a 600,000-Gaussian system with a global linear transform, despite the former system having one-eighth of the number of parameters in total.

## 3.6 Summary

In this chapter we described the role of covariance modelling in ASR systems. We described the basic process of estimating the parameters of covariance models, and explained the need for models to generalise to unseen test data, and for the covariance matrices to be well-conditioned, in addition to the practical requirements for memory usage and the computational costs of decoding.

We then described commonly-used methods for covariance modelling: constraining the matrices to be diagonal or block-diagonal; factor-analysed models; and precision matrix subspace methods. We explained our motivation for investigating full covariance models.

# Chapter 4

# Gaussian graphical modelling

Graphical models (Lauritzen, 1996) are a means of intuitively representing the dependencies present in multivariate data. In this chapter we explain how graphical modelling can be applied to multivariate Gaussian data, and how the assignment of a prescribed conditional dependency structure to acoustic feature vectors can be viewed as a covariance model, and describe how parameter estimation may be performed. We consider these models in the context of CD-HMM systems using the formalism of buried Markov models (Bilmes, 1999). Graphical model structure can be learnt from data, and we briefly review work in this area used for ASR.

We consider graphical modelling with reference to the covariance modelling criteria from Chapter 3, and introduce alternative structure learning and parameter estimation methods from other fields, which we later use experimentally for full covariance GMM acoustic modelling.

## 4.1 Graphical models introduced

### 4.1.1 Types of graphical model

A graphical model is a graph with vertices corresponding to individual random variables and edges corresponding to dependencies between variables. The variables may be observed or hidden, discrete or continuous. Hidden (or *latent*) variables may even be specially constructed in order to simplify the dependency structure between other variables. The graphs may be directed and acyclic, in which case variables can be ordered according to parent-child relationships, with the parent taken to have some causal effect on the child; or undirected, where relationships are symmetric. The former are also known as *Bayesian Networks*, the latter as *Markov Random Fields*. The HMM can be presented as a directed graphical model with variables indexed over time – known as a *dynamic* Bayesian network (DBN). This is illustrated in Figure 4.1. DBNs have been successfully used to implement explicit models for speech decoding (Bilmes & Bartels, 2005) without recourse to speech-specific software.

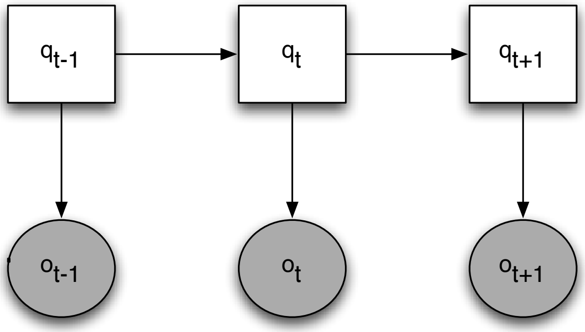

Figure 4.1: Graphical Model representation of a Hidden Markov model. By common convention discrete variables are represented by squares and continuous variables by circles; observed variables are shaded and latent variables are unshaded.

Bilmes (2000a) introduced dynamic Bayesian Multinets (DBMs). A Bayesian multinet is a graphical model, the structure of which is determined by the value of one or more of the discrete variables. These variables are known as "switching parents". If a series of Bayesian multinets, one for each frame of speech, are chained together, the resulting model is a DBM, also known as a Buried Markov Model (Bilmes, 1999). In this case, potential dependencies can extend over multiple frames. As used in (Bilmes, 2000a), the single switching parent in each frame is the hidden class variable of interest and all other variables are observed and continuous. The switching dependencies are implemented with a directed Gaussian graphical model, although an undirected graphical model could also be used. Both directed and undirected graphical models model *conditional* dependencies between variables. Their use for ASR is motivated by the theory that the acoustic features we wish to model have an underlying sparse dependency structure.

We can view a Buried Markov Model as a covariance model. The latent "switching parent" variable is the Gaussian index, $m$. Then the covariance matrix $U_m$ is constrained to satisfy the dependency structure specified by $m$. We can avoid the need to explicitly extend dependencies over multiple frames simply by extending each feature vector, $o_t$, as necessary. We explain this in more detail in the following sections, and go on to discuss how the dependency structure may be specified.

### 4.1.2 Gaussians as directed graphical models

Suppose we have a set of continuous random variables $X = (X_1, X_2, \ldots, X_d)$, with a continuous distribution[1] . A *directed graphical model* (DGM) for $X$ is a graph $\mathcal{G} = (V, E)$ with vertices $V = \{1, \ldots, d\}$ and directed edges $E \subseteq V \times V$. Each vertex $i \in V$ corresponds to variable $X_i$, and each edge $(i, j) \in E$, which we also write as $j \to i$, corresponds to causal relationships between the variables $X_i$ and $X_j$, with $X_j$ being the parent. The parents can be indexed by $pa(i) \subseteq \{1, \ldots, i-1\}$, with $X_{pa(i)} \equiv \{X_j : j \in pa(i)\}$. (Note that this ordering convention varies between authors.)

The dependencies are represented on the graph by a directed arc $j \to i$ indicating that $j \in pa(i)$. Of course the graph must be *acyclic*.

Given their parents, variables are conditionally independent of all other potential parents. This is known as the Markov property, and can be stated as:

$$X_i \perp\!\!\!\perp X_j \,|\, X_{pa(i)} \quad \text{for all } j < i, \quad j \notin pa(i) \tag{4.1}$$

Equivalently, the density function of $X$ can be factorised in terms of the conditional density functions as:

$$f_X(x) = \prod_{i=1}^{d} f(x_i | x_{pa(i)}) \tag{4.2}$$

[1] In the following sections it is helpful to distinguish between random variables, $X$, and samples, $x$

The idea is that parents should be *explanatory* variables for their children, with a relationship that can be explicitly represented in an expression for the distribution of the child.

A Gaussian DGM (see Pennoni, 2004, for example) is equivalent to a series of linear regression systems, with $X_i$ expressed as a linear function of its parents and a residual, the independent random variable $\epsilon_i \sim N(\mu_i, \sigma_i^2)$ Since the relationships are linear, we can write this as

$$X_i = \omega_{i1} X_1 + \omega_{i2} X_2 + \cdots + \omega_{i,i-1} X_{i-1} + \epsilon_i \tag{4.3}$$

where the $\omega_{ij}$ are the *partial regression coefficients* and are zero if $X_j$ is not a parent of $X_i$ $(j \notin pa(i))$. Alternatively, this can be expressed as

$$LX = \epsilon \tag{4.4}$$

where matrix $L$ is lower triangular, with diagonal elements all equal to one. Its lower-triangular elements consist of the negatives of the partial regression coefficients, $L_{ij} = -\omega_{ij}$. From this, we obtain

$$X = L^{-1}\epsilon \tag{4.5}$$

from which it can be seen that $X$ has a multivariate Gaussian distribution. Note that as $L$ is lower triangular, so is $L^{-1}$. Writing

$$\text{cov}(\epsilon) := \Lambda = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_k^2 \end{pmatrix} \tag{4.6}$$

we obtain

$$U = L^{-1}\Lambda(L^{-1})^T \tag{4.7}$$

for the covariance matrix of $X$.

This process can be reversed to obtain a DGM structure from a covariance matrix $U$, provided the indices of the matrix correspond to the desired variable ordering. This is because any positive definite symmetric matrix has a unique decomposition $LDL^T$, where $L$ is lower triangular with ones along the diagonal and $D$ is diagonal (this can be obtained from the Cholesky factorisation).

Furthermore, we see that the DGM can be represented in precision matrix form,

$$P = L^T \Lambda^{-1} L \tag{4.8}$$

It is clear that the matrices $L$, specifying the graphical model structure, could be shared over multiple Gaussians, with the parameters $\Lambda$ remaining specific to each individual Gaussian, similar to the semi-tied covariance matrix scheme discussed in Section 3.3.2.

### 4.1.3 Gaussians as undirected graphical models

Like a DGM, an *Undirected Graphical Model* (UGM) for the set of variables $X$ is a graph $\mathcal{G} = (V, E)$ as above, except that no variable ordering is required, and the arcs between variables are *undirected.* The *absence* of an arc between $i$ and $j$ indicates that $X_i$ and $X_j$ are conditionally independent, given all other variables in the system:

$$X_i \perp\!\!\!\perp X_j \,|\, X \backslash \{X_i, X_j\} \tag{4.9}$$

This is equivalent to it being possible to factorise the density function as:

$$f_X(x) = f(x_i, x_j | x_s) f(x_s) = f(x_i|x_s) f(x_j|x_s) f(x_s)$$

where $x_s = \{x_l : l \neq i, l \neq j\}$. Furthermore, writing $X_{\mathcal{A}} = \{X_i : i \in \mathcal{A}\}$ for any $\mathcal{A} \subseteq V$, we have the global Markov property

$$X_{\mathcal{A}} \perp\!\!\!\perp X_{\mathcal{B}} \,|\, X_{\mathcal{S}} \tag{4.10}$$

if $\mathcal{S}$ *separates* $\mathcal{A}$ from $\mathcal{B}$ in the graph – that is, all possible paths in $\mathcal{G}$ from elements of $\mathcal{A}$ to elements of $\mathcal{B}$ pass through $\mathcal{S}$. Equivalently, the distribution can be factorised according to the cliques, $\mathcal{C}$ of $\mathcal{G}$:

$$f_X(x) = \prod_{c \in \mathcal{C}} \psi(x_c) \tag{4.11}$$

A multivariate Gaussian with a sparse conditional dependency structure may be readily represented as a UGM by considering the zeros of the precision matrix $P = U^{-1}$. A non-zero element $P_{ij}$ in the precision matrix corresponds to the presence of an arc between vertices $i$ and $j$ in the graphical model. If the element is zero, there is no arc. This is a well-known result (see Lauritzen, 1996).

Imposing a UGM structure on a multivariate Gaussian can therefore be viewed as a precision matrix model. Such models – also known as *covariance selection* models – were first studied by Dempster (1972); many associated results were derived by Porteous (1985).

As an illustration (Jones & West, 2005), consider the density function of the system:

$$f_X(x;P) \propto \exp\{-\frac{1}{2}x^T P x\} \tag{4.12}$$

$$\Rightarrow f(x_i, x_j|x_s;P) \propto \exp\{-\frac{1}{2}(2x_i P_{ij} x_j + g_1(x_i,x_s) + g_2(x_j,x_s))\} \tag{4.13}$$

where $g_1(x_i, x_s)$ and $g_2(x_j, x_s)$ are just the other terms of the matrix multiplication, and again $x_s = \{x_l : l \neq i, l \neq j\}$. From this we can see that the conditional density can be factorised into $f(x_i|x_s)$ and $f(x_j|x_s)$ terms if and only if $P_{ij} = 0$.

## 4.2 Introduction to structure learning

As we have seen in Section 4.1, graphical modelling, applied to Gaussian systems via the formalism of Buried Markov Models, is equivalent to imposing a sparsity structure on each precision matrix, or some factorisation thereof. Bearing in mind the criteria for covariance modelling specified in Section 3.1, for GMs to be useful covariance models, we require our imposition of sparsity to result in learned matrices that are either better conditioned, or have improved generalisation ability, whilst retaining sufficient modelling power from the unconstrained case. GM covariance modelling can be divided into two sub-problems:

- Obtaining the optimal sparsity structure;
- Learning the optimal parameters for the desired structure.

However, as we shall show, the two problems may be solved simultaneously.

The principal advantage of the DGM representation introduced in Section 4.1.2 is that the parameters pertaining to dependency structure may be learned separately to the Gaussian-specific conditional variances, $\Lambda_m$. However, the usefulness of being able to impose sparsity on the matrices $L$ is then limited: by sharing parameters across multiple Gaussians in the manner of STC systems, we can control the amount of data available to estimate them. We therefore consider only the case where no parameters are shared across Gaussians.

As we have seen in Section 4.1.3, the natural DGM representation for speech can be converted to a UGM representation; so the structure learning problem is equivalent, considering the variables of interest as a multivariate Gaussian system, to selecting which elements of the precision matrix, $P_m$, should be set to zero. This is, of course, very different to selecting zeros of a sparse covariance matrix: for example, consider a model where the graph is a chain. In this case, the precision matrix would have a banded-diagonal structure, whereas the covariance matrix would have no zero elements. This can be contrasted with the approaches in Section 3.2, where instead a sparsity structure is imposed directly on the covariance matrix.

### 4.2.1 Structure learning with mutual information

Information theory has long been used for language modelling for ASR, and more recently attempts have been made to utilise it for acoustic modelling. The entropy, $H(X_i)$, of a random variable $X_i$ is a measure of how uncertain its outcome is, with $H(X_i)$ at a maximum for the uniform distribution. The mutual information between two variables $X_i$ and $X_j$ is a measure of the additional information gained about $X_i$ by observing $X_j$ or vice verse, and is given by

$$I(X_i; X_j) = H(X_i) - H(X_i|X_j) = H(X_j) - H(X_j|X_i) \tag{4.14}$$

where $H(X_i|X_j)$ is the conditional entropy of $X_i$, given $X_j$. For the discussion that follows we assume the variables $X_i$ to be continuous. The mutual information is given by

$$I(X_i; X_j) = \int f(x_i, x_j) \log \frac{f(x_i, x_j)}{f(x_i)f(x_j)} dx_i dx_j \tag{4.15}$$

Mutual information for feature selection has been used by Scanlon *et al.* (2003) for phone classification: elements $X_i$ from a time-frequency space grid are selected as input to a neural network classifier if the $I(X_i; Y)$ are sufficiently high, with $Y$ being the class-label of interest.

Bilmes (1998) used mutual information for modelling the joint distribution of features from a time-frequency grid, where, for a feature variable $X_i$, the correlations between $X_i$ and possible dependencies $X_j$ are modelled if $I(X_i; X_j)$ is sufficiently high. This was extended (Bilmes, 2000b; Bilmes *et al.*, 2001; Ellis & Bilmes, 2000) to condition the mutual information on the class of interest, $Y$. This is known as *conditional* mutual information (CMI), given by

$$I(X_i; X_j|Y) = \sum_y \int f(x_i, x_j, y) \log \frac{f(x_i, x_j|y)}{f(x_i|y)f(x_j|y)} dx_i dx_j \tag{4.16}$$

Further to this, Bilmes (2000a); Zweig *et al.* (2002) introduce a discriminative mutual information based measure for dependency selection in the context of DBMs (where the $X_j$ are the parents in the graphical model) called the *explaining away residual* or *EAR measure*, given by

$$EAR(X_i, X_j) = I(X_i; X_j|Y) - I(X_i; X_j) \tag{4.17}$$

and show that choosing dependencies $X_J = \cup_j X_j$ to maximise this measure will maximise the expected class posterior probability $\mathbb{E}[P(Y|X_i, X_j)]$ for a fixed number of dependencies.

There are a number of drawbacks to this approach: firstly it is not practical to compute the globally optimal $X_J$ directly, and so variables $X_j$ are selected one at a time using a greedy search based on the EAR measure. This means that it is necessary to check that each additional parent adds class-conditional information above that provided by existing parents – this will be the case if $I(X_i; X_j|Y)$ is high relative to $I(X_{pa}; X_i|Y)$ where $X_{pa}$ is the set of parents already selected.

Secondly, in the usual case where the system is assumed to be a DGM, the mutual information between single variables $X_i$, $X_j$ is a monotonically increasing function of their correlation, $\rho_{x_i x_j}$, given by

$$I(X_i; X_j) = -\frac{1}{2} \log(1 - \rho^2_{x_i x_j}) \tag{4.18}$$

so computing the mutual information from data is essentially equivalent to obtaining estimates of the correlation coefficients. The method is not robust to limited data, high-dimension situations where the sample covariance matrix is a poor estimate of the true matrix.

Also note that the approaches described here ignore the advantage of the DGM/UGM formulation, that a sparse conditional independence structure, such as that determined by the zeros of the precision matrix, may be a more natural representation than a sparse marginal independence structure, determined by the zeros of the covariance matrix, or equivalently zeros of mutual information.

## 4.3 Parameter estimation

### 4.3.1 Basic methodology

Recall once more that our problem of determining graphical model structure within the covariance selection framework is equivalent to fixing the zeros of each precision matrix, $P_m$ (We drop the dependency on $m$ in what follows). The problem of how to determine this matrix from data was first considered by Dempster (1972). We assume for the moment that the graph structure is fixed, and there is only one class to consider.

Suppose we have a sample covariance matrix $S$, and assume an undirected graph $\mathcal{G} = (V, E)$ representing the structure of the multivariate Gaussian data. It is wished to obtain an optimal estimate of the covariance matrix, $\hat{U}$, or its inverse, $\hat{P}$, corresponding to this graphical model.

Dempster proposed the following rules:

- $\hat{P}$ should match the graph structure: set $\hat{P}_{ij} = 0$ for $(i, j) \notin E$
- $\hat{U}$ should agree with $S$ as much a possible: set $\hat{U}_{ij} = S_{ij}$ for $(i, j) \in E$

Dempster showed that a covariance matrix chosen according to the above has the following attractive properties:

1. Assuming that $S$ itself is positive definite, a unique $\hat{U}$ always exists and is positive definite
2. Of all possible Gaussian models such that $U_{ij} = S_{ij}$ for $(i, j) \in E$, the choice $\hat{U}$ is the maximum entropy model, often considered optimal for prediction.
3. Of all possible Gaussian models such that $P_{ij} = 0$ for $(i, j) \notin E$, the choice $\hat{P}$ has maximum likelihood.

The $\hat{U}$ or $\hat{P}$ specified by Dempster's theory cannot be computed directly. In (Dempster, 1972) an iterative procedure is described, based on the theory of exponential distributions, using the Newton-Raphson method. The solution converges quickly; however, each iteration requires $O(n^2)$ computations, where $n$ is the number of free parameters in $P$. Dahl *et al.* (year unknown) details how the computations can be made more efficient by means of *triangulating* $\mathcal{G}$ (equivalently referred to as finding a *chordal embedding* of $\mathcal{G}$).

### 4.3.2 Estimation as an optimisation problem

We now approach the estimation problem from an alternative direction, demonstrating the parallels with Dempster's work. From equation 3.7, we have:

$$\ell(P) := \log f(o\ ;\ P) = \frac{\beta}{2}(\log|P| - \operatorname{tr} PS) \tag{4.19}$$

and the problem of finding $P$ to maximise the likelihood, subject to the constraints of a given graphical model, can be expressed as a convex optimisation problem:

$$\begin{aligned} \text{minimise} \quad & -\log|P| + \operatorname{tr} PS \\ \text{subject to} \quad & P \succeq 0 \\ & P_{ij} = 0 \quad (i,j) \notin E \end{aligned} \tag{4.20}$$

This type of problem, where $P$ is symmetric, is known as a *semidefinite program*. The first constraint expresses the requirement that $P$ be positive semidefinite[1]. We introduce the dual variable $\Theta$ given by

$$[\Theta]_{ij} = \begin{cases} 0 \ (i,j) \in E \\ \theta_{ij} \text{ otherwise} \end{cases} \tag{4.21}$$

This definition allows us to readily express the equality constraints of the optimisation problem in the Lagrangian dual. This is given by

$$g(\Theta) = \inf_{P \succeq 0} \{-\log|P| + \operatorname{tr} PS + \operatorname{tr} \Theta P\} \tag{4.22}$$

The last two terms can be combined into one by making a changing of variables. We define

$$[\Phi]_{ij} = \begin{cases} S_{ij} \ (i,j) \in E \\ \varphi_{ij} \text{ otherwise} \end{cases} \tag{4.23}$$

[1] In practice we require $P$ to be positive definite for use in our models; the relaxation here ensures that a solution exists to the optimisation problem

where $\varphi_{ij} = \theta_{ij} + S_{ij}$. (The $S_{ij}$ are of course fixed). The Lagrangian becomes

$$g(\Phi) = \inf_{P \succeq 0} \{\operatorname{tr} \Phi P - \log |P|\} \tag{4.24}$$

To find the minimum, we differentiate with respect to $P$ and set the derivative equal to zero, obtaining

$$\Phi = P^{-1} = U \tag{4.25}$$

noting that the requirement that $P$ is positive definite implies that its inverse must also be. Substituting this into the dual function we have

$$g(\Phi) = d + \log |\Phi| \tag{4.26}$$

so the dual optimisation problem is:

$$\begin{aligned} \text{maximise} \quad & \log |\Phi| \\ \text{subject to} \quad & \Phi \succeq 0 \end{aligned} \tag{4.27}$$

or equivalently

$$\begin{aligned} \text{maximise} \quad & \log |U| \\ \text{subject to} \quad & U \succeq 0 \\ & U_{ij} = S_{ij} \quad (i,j) \in E \end{aligned} \tag{4.28}$$

This is known as a maximum-determinant positive-definite matrix completion problem. We can see that problem of finding $P$ to maximise the likelihood, subject to the constraints of the graphical model, can be converted to the problem of finding the $U$ with maximum determinant, subject to constraints based on the sample covariance matrix, being the same constraints obtained by Dempster.

## 4.4 Penalised likelihood method of GM structure learning

### 4.4.1 Background

We consider how the optimal set of edges of $\mathcal{G}$ may be determined for the covariance selection models. As discussed at the beginning of this chapter, our goal is to select dependencies that result in models with good modelling power that are robust when data is limited. This is different to many applications of the models, including Dempster (1972), where it may be more important to determine only and all the dependencies which truly do exist in data – an example might be the functional grouping of genes in Bioinformatics. In this case it would be natural to infer a dependency only if the data provide statistically significant evidence for it – this is not appropriate here.

We cannot simply select the edges which give the maximum likelihood of the data, since this will be at a maximum when the number of parameters are at a maximum – when the graph is complete. A solution to this issue would be instead to maximise a penalised version of the likelihood where the penalty is related to the number of edges, learning the parameters simultaneously with the graph structure. For example:

$$\hat{P} = \arg\max_{P} \{\ell(P) - \beta\lambda|E(P)|\} \tag{4.29}$$

Note that $|E|$, the number of edges of the graph, is equal to the number of non-zero off-diagonal elements of $\Omega$. The method is similar to that of choosing a set of edges of a fixed size in such a way as to maximise the likelihood. The problem in both cases is typically solved (approximately) in a similar manner to the EAR-measure problem in Section 4.2.1, by means of a greedy search. This is computationally intensive, and will not necessarily result in an optimal solution.

An alternative is to replace the penalty with the sum of the magnitude of the off-diagonal elements. The use of this penalty term was proposed by Tibshirani (1996) for regression and is known as the Lasso. The objective function, expressed as

$$\hat{P} = \arg\max_{P} \{\ell(P) - \beta\nu \sum_{i \neq j} |P_{ij}|\} \tag{4.30}$$

is a convex optimisation problem, which can be efficiently solved. The use of the Lasso has been used for graphical model structure learning by Meinshausen & Bühlman (2006), selecting the neighbours of each variable in the model: but this method does not ensure that the final matrix is well-conditioned. Yuan & Lin (2007) use a similar method, but explicitly ensure that the resulting matrices are positive definite.

For the graphical modelling work in this thesis, we adopt an alternative approach, that of Banerjee *et al.* (2006), which we describe in detail in the following section. The technique adopts an alternative penalty term equal to the sum of the magnitude of *all* elements of $P$, and the optimisation guarantees bounds on the eigenvalues of the resulting matrix.

### 4.4.2 Penalised likelihood with the $l_1$ norm

We define the $l_q$ norm of a matrix $A$ by[1]

$$\|A\|_q = (\sum_i^d \sum_j^d |A_{ij}|^q)^{\frac{1}{q}} \tag{4.31}$$

for $q \geq 1$. This an example of an *entry-wise norm.* In the case $q = \infty$, the largest term in the sum dominates, so this is the *maximum value norm.*

Banerjee *et al.* (2006) proposed using the $l_1$ norm of the matrix $P$ as a penalty term for the likelihood. We maximise

$$\ell(P) - \rho\beta\|P\|_1 \tag{4.32}$$

We show in Appendix A.3.4 that $q = 1$ is the unique choice for which the resulting $\hat{P}$ is a sparse matrix. The optimisation problem (4.20) becomes:

$$\begin{aligned} \text{minimise} \quad & \rho\|P\|_1 - \log|P| + \operatorname{tr} PS \\ \text{subject to} \quad & P \succ 0 \end{aligned} \tag{4.33}$$

The parameter $\rho > 0$ controls the size of the penalty, and hence the sparsity of the solution. In Appendix A.3.3, we show that the problem may be solved via its dual:

$$\begin{aligned} \text{maximise} \quad & d + \log|U| \\ \text{subject to} \quad & \|U - S\|_\infty \leq \rho \\ & U \succ 0 \end{aligned} \tag{4.34}$$

Banerjee *et al.* (2006) show that for any $\rho > 0$, the solution is bounded as follows:

$$aI \preceq \hat{P} \preceq bI \tag{4.35}$$

where

$$a = \frac{1}{\|S\|_{SV} + d\rho}, \quad b = \frac{d}{\rho} \tag{4.36}$$

These bounds are equivalent to imposing bounds on the largest and smallest eigenvalues of $U$, and hence on its condition number.

[1] The notation $\|.\|_q$, which we adopt from Banerjee *et al.* (2006), is not standard – it may also be used to denote the *operator norm* of a matrix.

### 4.4.3 An algorithm

We now describe the algorithm used by Banerjee *et al.* (2006) for solving the dual problem. The idea is to maximise $|U|$ by optimising over one row and column at a time. The diagonal elements of the solution, $\hat{U}_{ii}$ will be set to $S_{ii} + \rho$, their maximum value under the constraints imposed by the $l_\infty$ norm: we therefore initialise $U^0 = S + \rho I$. Since $S \succeq 0$, $U \succ 0$ for any $\rho > 0$. Also note that after removing any row/column pair from $U$, the resulting matrix is still positive definite (this follows directly from the definition, by choosing the vector appropriately).

To optimise over the $i^{\text{th}}$ row and column the matrix is permuted so that they are the last row and column. This does not change the determinant. Then we partition $U$ as

$$U = \begin{pmatrix} U_{11} & u_{12} \\ u_{12}^T & u_{22} \end{pmatrix} \tag{4.37}$$

We have $U_{11} \in \mathbb{S}_{++}^{d-1}$, $u_{12} \in \mathbb{R}^{d-1}$. As for all diagonal elements, $u_{22}$ is fixed. We wish to find the vector $u_{12}$ maximising the determinant, subject to the constraint $\|U - S\|_\infty \leq \rho$, which translates as $\|u_{12} - s_{12}\|_\infty \leq \rho$. Taking the Schur complement[1] of $U_{11}$, we obtain

$$\det U = \det U_{11} \det(u_{22} - u_{12}^T U_{11}^{-1} u_{12}) \tag{4.38}$$

and the maximisation problem becomes

$$\hat{u}_{12} = \arg\min \{u^T U_{11}^{-1} u \,|\, \|u - s_{12}\|_\infty \leq \rho\} \tag{4.39}$$

which can be solved by standard off-the-shelf Quadratic Programming algorithms.

After every iteration, the matrix $U$ is positive definite. The convergence of the algorithm can be checked after each cycle through the rows/columns by computing the difference between the values of the primal and the dual,

$$f(P^{(n)}) - g(U^{(n)}) = \rho \|P^{(n)}\|_1 + \operatorname{tr} P^{(n)} S - d \tag{4.40}$$

[1] See Appendix A.3

### 4.4.4 Practical issues

In a practical implementation of the penalised likelihood method for sparse GM learning, a number of practical issues must be considered. Choosing the penalty parameter $\rho$ is important. Meinshausen (2005) showed that the estimator is not consistent as the quantity of training data tends to infinity, if $\rho$ is fixed. Assuming that optimal model estimation, rather than a restriction on the number of parameters, is the primary objective, it is clear that as $\beta \to \infty$ we should have $\rho \to 0$ to recover the optimality of the sample covariance matrix in this case. Banerjee *et al.* (2006) suggest a heuristic for $\rho$ which approximately yields $\rho \propto \beta^{-1/2}$, proportional to the estimator variance. This would mean setting the parameter on a per-Gaussian basis.

A second issue is the fact that the method as presented is not invariant to arbitrary feature scaling. Since it is the precision matrix in the penalty term, the logical option would be to scale the system so that the diagonal elements of the precision matrix (corresponding to the conditional correlation coefficients) are all one. However, this is not possible, since a good estimate of the precision matrix is not known in advance in the case when the sample covariance matrix is poorly conditioned.

## 4.5 Summary

In this chapter we introduced Gaussian graphical modelling as a form of covariance modelling. We explained the correspondence between the dependency structure of the graphical model and the sparsity structure of the precision matrix. We described methods used for learning the structure of graphical models used for ASR, and discussed parameter estimation using the covariance selection framework. Finally, we introduced a technique for learning the model parameters and structure simultaneously by maximising a penalised-likelihood function. We carry out ASR experiments using this technique the following chapter.

# Chapter 5

# The shrinkage estimator

In this chapter we introduce the "shrinkage estimator", an alternative to the maximum likelihood estimator for full covariance modelling. We adopt a generative approach here, and broadly follow the analysis of Ledoit & Wolf (2004). Whilst the maximum likelihood estimator (MLE) has several attractive asymptotic properties, they argue that weaker asymptotic assumptions are more appropriate, under which consistency of the MLE does not hold. The shrinkage estimator corrects for this, and explicitly improves upon the sample covariance matrix in terms of generalisation ability and conditioning.

The shrinkage estimator has been employed for covariance estimation in high-dimensionality situations when the number of matrices to estimate is relatively small, for example, portfolio selection in financial modelling (Ledoit & Wolf, 2003) and gene association in Bioinformatics (Schäfer & Strimmer, 2005). In this work we extend it for use in large-scale ASR systems. In particular, we adapt the methods for use in GMMs with many thousands of Gaussians. We show how the shrinkage estimator can be related to Bayesian techniques that have been successfully used for covariance smoothing in ASR (Chen *et al.*, 2006).

## 5.1 Shrinkage introduced

### 5.1.1 Generative approach

In Chapter 3 we were somewhat non-committal about what precisely was meant by an optimal covariance matrix $U^*$. In this chapter we generally adopt a classical statistical approach, using a generative model. That is to say, we assume that our statistical model is correct, and moreover, that the model parameters, whilst unknown, have fixed, true values. All expectations are taken with respect to this true distribution. Throughout this chapter we drop dependence on the Gaussian $m$ for simplicity of presentation, and denote the true matrix by $\Sigma$. Most of the analysis below was derived by Ledoit & Wolf (2003) for situations where there are no hidden variables; we describe our method for removing this requirement.

Within this generative framework, the performance of an estimator may be measured by its expected squared deviation from the true parameter, the *mean squared error* (MSE):

$$\mathrm{MSE}(U) = \mathbb{E}\|U - \Sigma\|^2 \tag{5.1}$$

In this respect, the maximum likelihood estimator (MLE) has several attractive properties. Firstly, it is *consistent*: the MLE of a parameter $\theta$, based on $n$ samples, converges (in probability) to $\theta$ as $n \to \infty$. Secondly, it is *asymptotically efficient*: as $n \to \infty$, the MLE variance converges to the minimum variance possible for any unbiased estimator, given by the Cramer Rao lower bound. So that if $n$ is large, the variance of $S^n$ is close to the minimum achievable. The consistency of the MLE tells us that the sample covariance matrix based on $n$ samples, $S^n$ obeys

$$\lim_{n \to \infty} \mathbb{E}\|S^n - \Sigma\| = 0 \tag{5.2}$$

whilst the asymptotic efficiency property tells us that if $n$ is large, the variance of $S^n$ is close to the minimum achievable for that value of $n$.

From these results it would appear that $S^n$ is a good choice of estimator. However, rarely can we consider the amount of training data to be approaching infinity, so it is questionable whether these results are useful. An alternative is suggested in Section 5.1.2 below.

### The Frobenius norm

To measure the error of a matrix estimator, we use the Frobenius norm[1],

$$\|A\|_F = (\sum_i \sum_j |A_{ij}|^2)^{\frac{1}{2}} = \sqrt{\operatorname{tr} A^T A} \tag{5.3}$$

[1] Following the notation of Chapter 4, we could denote this by $\|.\|_2$, but here we follow the more standard notation.

This arises from the inner product $\langle A, B\rangle_F = \operatorname{tr} A^T B$. In the equations that follow, the Frobenius norm and corresponding inner product are used implicitly. An important property of the Frobenius norm is that it is invariant to rotation:

$$\begin{aligned} \|R^T A R\|^2 &= \operatorname{tr}(R^T A R)^T (R^T A R) && (5.4)\\ &= \operatorname{tr} R^T A^T R R^T A R && (5.5)\\ &= \operatorname{tr} A^T A R R^T = \operatorname{tr} A^T A = \|A\|^2 && (5.6) \end{aligned}$$

where we use the fact that $R^T R = I$, for any rotation $R$, and the fact that the trace operator is invariant to cyclic permutations.

### 5.1.2 Covariance estimation with weak asymptotic assumptions

Ledoit & Wolf (2004) note that the usual $n \to \infty$ assumption underpinning MLE is not appropriate in the situation where the number of parameters is large relative to the number of samples. They analyse the sample covariance matrix under a weaker set of assumptions which they term *general asymptotics*. The principal of these is that the ratio $d/n$ is bounded, so that while $n$ may grow to infinity, it does not grow faster than the dimensionality $d$. This may be more reasonable for covariance modelling for ASR, where, as the amount of training data grows, we may extend the feature vector (using windowing, for example), increase the number of Gaussians, or split the state space to include more context.

We summarise some of the technical results from Ledoit & Wolf (2004) using the notation introduced in Section 3.1.2. For simplicity here we set the mean to zero and take $x^{(k)}$ to be random samples from the distribution. The sample covariance matrix is given by

$$S^n = \frac{1}{n} \sum_k^n x^{(k)} x^{(k)T} \qquad (5.7)$$

We decompose the true covariance matrix as

$$\Sigma = \Gamma \Lambda \Gamma^T \qquad (5.8)$$

where $\Lambda$ is diagonal. Then $z^{(k)} = \Gamma^T x^{(k)}$ are samples for which all elements are uncorrelated, with $\mathbb{E}(z^{(k)} z^{(k)T}) = \Lambda$

In this analysis, to account for the fact that $d$ is not held constant, we normalise the error by $d$. If there are $n$ samples, the mean squared error is given by

$$\begin{aligned}\frac{1}{d}\mathbb{E}\|S^n - \Sigma\|^2 &= \frac{1}{d}\mathbb{E}\|\Gamma^T S^n \Gamma - \Lambda\|^2 &(5.9)\\ &= \frac{1}{d}\mathbb{E}\left\|\frac{1}{n}\sum_k^n z^{(k)} z^{(k)T} - \Lambda\right\|^2 &(5.10)\\ &= \frac{1}{dn}\mathbb{E}\|zz^T - \Lambda\|^2 &(5.11)\\ &= \frac{1}{dn}\sum_i^d\sum_j^d \mathbb{E}(z_i^2 z_j^2) - \frac{1}{dn}\sum_i^d \lambda_i^2 &(5.12)\end{aligned}$$

where $z$ is an arbitrary uncorrelated sample from the distribution. The first step uses the fact that the Frobenius norm is invariant to rotation, and (5.11) uses the fact that the samples are uncorrelated. This expression is dominated by the first term. Using the fact that the the elements of $z$ are uncorrelated, we can re-express this as:

$$\begin{aligned}\frac{1}{dn}\sum_i^d\sum_j^d \mathbb{E}(z_i^2 z_j^2) &= \frac{d}{n}\mathbb{E}\left(\frac{1}{d}\sum_i^d z_i^2\right)^2 &(5.13)\\ &= \frac{d}{n}\left(\mathbb{E}\frac{1}{d}\sum_i^d z_i^2\right)^2 + \frac{d}{n}\operatorname{var}\left(\frac{1}{d}\sum_i^d z_i^2\right) &(5.14)\\ &= \frac{d}{n}\bar{\lambda}^2 + \frac{d}{n}\operatorname{var}\left(\frac{1}{d}\sum_i^d z_i^2\right) &(5.15)\end{aligned}$$

So we see that the MSE does not generally vanish for bounded $d/n$. This analysis motivates an alternative choice of covariance estimator. We aim to find one which can consistently minimise the MSE under these weaker asymptotic assumptions.

### 5.1.3 Shrinkage: the bias-variance trade-off

Stein (1956) first introduced the concept of “shrinkage” as applied to high-dimensional estimators (specifically, of the mean of a distribution), deriving the surprising result that the performance of the MLE can always be improved upon by shrinking by a given factor $\alpha$ (the “shrinkage intensity”) towards some central value. More recently, Ledoit & Wolf (2004) showed how this procedure can be applied to covariance matrices. The shrinkage estimator of $\Sigma$, is given by

$$U = (1 - \alpha)S + \alpha D \tag{5.16}$$

where $D$, the "shrinkage target", is a diagonal matrix. It can be seen that as $\alpha$ is increased to one, the off-diagonal elements of $U$ shrink towards zero.

The estimator MSE, introduced in Equation 5.1 can be decomposed into variance and bias terms as follows:

$$\begin{aligned} \mathbb{E}\|U - \Sigma\|^2 &= \mathbb{E}\|(U - \mathbb{E}U) + (\mathbb{E}U - \Sigma)\|^2 && (5.17) \\ &= \mathbb{E}\|U - \mathbb{E}U\|^2 + \|\mathbb{E}U - \Sigma\|^2 && (5.18) \\ &= \mathrm{var}(U) + \mathrm{bias}^2(U) && (5.19) \end{aligned}$$

Typically, a higher dimensional estimator will have a lower bias, but higher variance – minimising the MSE of the shrinkage estimator can be viewed as optimising the trade-off between the two.

$S$ is an unbiased estimator of $\Sigma$, whilst $D$ is biased in its off-diagonal elements. The shrinkage procedure can therefore be viewed as "backing off" from the high-variance, unbiased $S$ to the low-variance, biased $D$. This is illustrated in Figure 5.1. It will be seen below that the optimal shrinkage factor $\alpha$ can be obtained analytically.

In Ledoit & Wolf (2004), $D$ is taken to be a uniform diagonal matrix $D = \rho I$. However, Schäfer & Strimmer (2005) discuss a variety of alternative targets. As they explain, the case where $D$ consists of the diagonal elements of $S$ is attractive: it preserves the diagonal elements of the matrix and makes it easy to estimate an optimal $\alpha$ in a scale-free manner. It is this target which we use throughout this work.

### 5.1.4 Bayesian interpretation

It is also possible to obtain a shrinkage-style estimator using a Bayesian approach. The MSE of an estimator $U$ can be replaced by the Bayes' risk with a quadratic loss function which is minimised by setting $U$ to the posterior mean. If a non-informative prior is chosen, we obtain the minimum risk at $U = S$ as in the

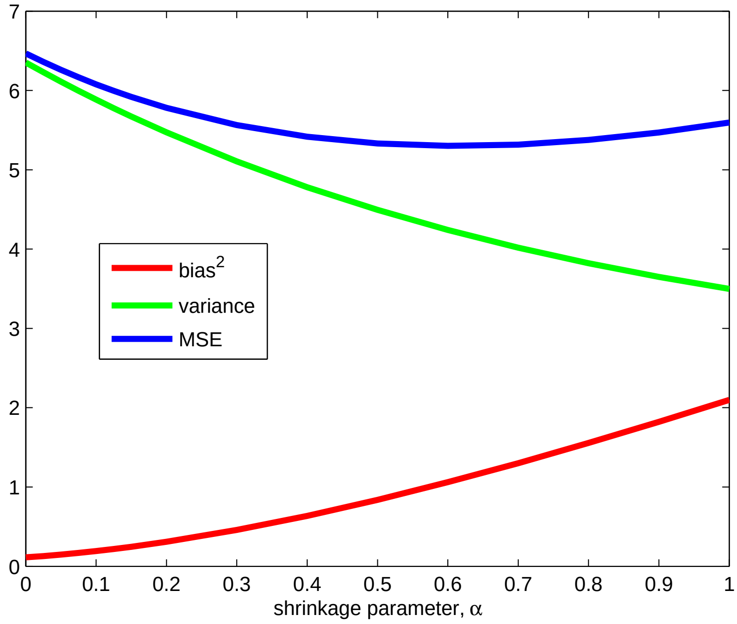

Figure 5.1: An illustration of the bias-variance trade-off with varying shrinkage intensity, using simulated data (taken from the example in Section 3.1)

classical MLE case. To obtain the shrinkage estimator, we use a conjugate prior to the multivariate Gaussian, the inverse-Wishart distribution:

$$p(\Sigma) = \mathcal{W}^{-1}(\tau + d + 1, \tau D) \tag{5.20}$$

where $D$ is the shrinkage target, and the first hyperparameter, $\tau$, reflects the strength of the prior (we refer to $\tau$ as the prior weight). We use $\beta$ to represent the number of training samples, for consistency with later sections. Setting $U$ to the posterior mean gives

$$\begin{aligned} U &= \frac{\beta S + \tau D}{(\beta + \tau + d + 1) - d - 1} && (5.21) \\ &= \frac{S}{\beta + \tau} + \frac{\tau D}{\beta + \tau} && (5.22) \\ &= (1 - \alpha) S + \alpha D && (5.23) \end{aligned}$$

for suitably chosen $\alpha$. (5.22) has the obvious Bayesian interpretation, that as the amount of training data available is reduced, resulting in a small value for $\beta$, the influence of the prior is increased, and the minimum Bayes' risk estimator becomes closer to $D$. This form of off-diagonal smoothing was mentioned briefly by Povey (2003) and further in Povey (2006); it was used in the IBM full covariance system (Chen *et al.*, 2006).

In the Bayesian interpretation, $\tau$ here is a constant that must be manually specified. In the references above, it was set to 100 or 200. In Section 5.2.6 we consider whether the analytically obtained shrinkage parameter can be expressed using such a constant, and later compare the two approaches experimentally.

### 5.1.5 Matrix conditioning

We discussed matrix conditioning in Chapter 3. We showed that when applying a rotation $R$ to symmetric matrix $A$, the most dispersed diagonal elements that can be obtained are the eigenvalues of $A$. This was used to show that the sample covariance matrix $S$ has eigenvalues that are, on average, more dispersed than those of the true matrix $\Sigma$, and also that the diagonal elements of the sample matrix are less dispersed.

We now consider the dispersion of the eigenvalues of the shrinkage estimator. We denote the eigenvalues by $l_i$. Note that the expected mean of the eigenvalues is given by

$$\mathbb{E}\frac{1}{d}\operatorname{tr} U = \mathbb{E}\frac{1}{d}\operatorname{tr} S = \bar{\lambda} \tag{5.24}$$

From (A.13) the expected dispersion is obtained from

$$\mathbb{E}\|U - \bar{\lambda} I\|^2 = \mathbb{E}\operatorname{tr}(U - \bar{\lambda} I)^2 = \mathbb{E}\sum_i^d (l_i - \bar{\lambda})^2 \tag{5.25}$$

This can be expressed as a weighted sum of the dispersion of the eigenvalues of $S$ and $D$:

$$\mathbb{E}\|U - \bar{\lambda} I\|^2 = (1-\alpha)^2\mathbb{E}\|S - \bar{\lambda} I\|^2 + (1-(1-\alpha)^2)\mathbb{E}\|D - \bar{\lambda} I\|^2 \tag{5.26}$$

Since the eigenvalues of $D$ are always less dispersed than those of $S$, this shows that eigenvalues of the shrinkage estimator are also less dispersed that the eigenvalues of $S$.

## 5.2 The shrinkage parameter

### 5.2.1 Optimising the shrinkage parameter

Ledoit & Wolf (2004) obtained a method for computing the optimal shrinkage intensity analytically, whilst Schäfer & Strimmer (2005) generalised this to a variety of shrinkage targets. We seek $\alpha$ to minimise

$$\begin{aligned}\mathbb{E}\|U - \Sigma\|^2 &= \mathbb{E}\|\alpha(D-\Sigma) + (1-\alpha)(S-\Sigma)\|^2 && (5.27)\\ &= \alpha^2\mathbb{E}\|D-\Sigma\|^2 + (1-\alpha)^2\mathbb{E}\|S-\Sigma\|^2 \\ &\quad + 2\alpha(1-\alpha)\mathbb{E}\langle D-\Sigma, S-\Sigma\rangle && (5.28)\end{aligned}$$

The derivative with respect to $\alpha$ is given by

$$\begin{aligned}\frac{d}{d\alpha}\mathbb{E}\|U-\Sigma\|^2 &= 2\alpha\mathbb{E}\|D-\Sigma\|^2 + (\alpha-1)\mathbb{E}\|S-\Sigma\|^2 \\ &\quad + 2(1-2\alpha)\mathbb{E}\langle D-\Sigma, S-\Sigma\rangle\end{aligned} \tag{5.29}$$

Setting this equal to zero, we obtain

$$\begin{aligned}&\mathbb{E}\|S-\Sigma\|^2 - \mathbb{E}\langle D-\Sigma, S-\Sigma\rangle \\ &\quad = \alpha[\mathbb{E}\|D-\Sigma\|^2 + \mathbb{E}\|S-\Sigma\|^2 - 2\mathbb{E}\langle D-\Sigma, S-\Sigma\rangle] && (5.30)\\ &\quad = \alpha\mathbb{E}\|(S-\Sigma) - (D-\Sigma)\|^2 && (5.31)\end{aligned}$$

We decompose $\Sigma$ into its diagonal and off-diagonal elements: $\Sigma = \Sigma^{\mathsf{diag}} + \Sigma^{\mathsf{od}}$. Since $\mathbb{E}S = \Sigma$, $\mathbb{E}\langle \Sigma^{\mathsf{od}}, S - \Sigma\rangle = 0$. We add this to the second term on the left-hand side, giving

$$\mathbb{E}\langle D - \Sigma^{\mathsf{diag}}, S - \Sigma\rangle = \mathbb{E}\|D - \Sigma^{\mathsf{diag}}\|^2 \tag{5.32}$$

since the off-diagonal terms then vanish from the inner product. We therefore obtain

$$\hat{\alpha} = \frac{\mathbb{E}\|S - \Sigma\|^2 - \mathbb{E}\|D - \Sigma^{\mathsf{diag}}\|^2}{\mathbb{E}\|S - D\|^2} \tag{5.33}$$

When $D$ consists simply of the diagonal elements of $S$, then the numerator in (5.33) becomes

$$\sum_{i\neq j} \mathbb{E}(S_{ij} - \Sigma_{ij})^2 = \sum_{i\neq j} \operatorname{var} S_{ij} \tag{5.34}$$

(since $S$ is unbiased) whilst the denominator becomes

$$\sum_{i\neq j} \mathbb{E}S_{ij}^2 \tag{5.35}$$

The numerator can be recognised as the off-diagonal elements of the matrix $\operatorname{var}(S)$. From (5.33) it can be seen that $\alpha$ increases with this variance, so that when the sample matrix has higher variance, the shrinkage target, $D$, achieves more prominence, as we would expect.

As presented, the calculations are not invariant to arbitrary scaling of feature dimensions. To remedy this we adopt the approach of Schäfer & Strimmer (2005), dividing each element $S_{ij}$ by $\sqrt{S_{ii}S_{jj}}$. Note that due to the choice of shrinkage target, the diagonal elements themselves are not changed by the smoothing process.

### 5.2.2 Shrinkage parameter estimation

Note that neither the numerator (5.34) nor the denominator (5.35) terms above can be obtained directly, and must themselves be estimated from the training data. We define

$$w_{ij}^{(k)} = x_i^{(k)} x_j^{(k)} \tag{5.36}$$

from which we can estimate

$$S_{ij} = \bar{w}_{ij} = \frac{1}{n}\sum_{k}^{n} w_{ij}^{(k)} \tag{5.37}$$

We can treat the $w_{ij}$ as IID random variables, with an sample variance given by

$$\widetilde{\text{var}}\, w_{ij} = \frac{1}{n-1}\sum_{k}^{n}(w_{ij}^{(k)} - S_{ij})^2 \tag{5.38}$$

Using the IID assumption, we can then estimate the variance of $S_{ij}$ by

$$\begin{aligned}\widetilde{\text{var}}\, S_{ij} &= \frac{1}{n^2}\sum_{k}^{n}\widetilde{\text{var}}\, w_{ij}^{(k)} &(5.39)\\ &= \frac{1}{n^2}.n.\widetilde{\text{var}}\, w_{ij} &(5.40)\\ &= \frac{1}{n(n-1)}\sum_{k}^{n}(w_{ij}^{(k)} - S_{ij})^2 &(5.41)\end{aligned}$$

The expectation in the denominator term (5.35) may simply be replaced with the sample equivalent. We can use these estimates to obtain an estimate of the optimal shrinkage parameter, $\tilde{\alpha}$, say. If we use $U(\hat{\alpha})$ to denote the shrinkage estimator obtained using the true optimal parameter, and $U(\tilde{\alpha})$ to denote its counterpart using $\tilde{\alpha}$, then an important result proved by Ledoit & Wolf (2004) is that

$$\mathbb{E}\|U(\tilde{\alpha}) - U(\hat{\alpha})\|^2 \rightarrow 0 \tag{5.42}$$

under the weaker asymptotic assumption that $d/n$ is bounded, but does not necessarily vanish, as $n \rightarrow \infty$. Effectively this means that it is easier to find a consistent shrinkage estimator than a consistent estimator of the covariance matrix itself. In the following sections we show how these results can be practically applied to an CD-HMM system.

### 5.2.3 The shrinkage parameter for an CD-HMM system

We now explain how the sample variance of $S_{ij}$ can be obtained within the context of the EM algorithm. In this analysis, we fix the number and weighting of observations for each Gaussian (i.e. the $\gamma(t)$ and $\beta$), but assume that the actual observations vary randomly according to the true distribution and are IID. This allows us to obtain an estimate of var $S_{ij}$ by adapting the formulae of Section 5.2.2 to take account of the weights $\gamma(t)$. We redefine

$$w_{ij}(t) = (o_i(t) - \hat{\mu}_i)(o_j(t) - \hat{\mu}_j) \tag{5.43}$$

The sample covariance matrix $S$ is the sample mean of these observations:

$$S_{ij} = \frac{\sum_t \gamma(t) w_{ij}(t)}{\beta} \tag{5.44}$$

It should be noted that the all estimates of variance presented here are slightly biased. This could be remedied by applying a correction factor at each stage given by

$$\frac{\beta^2}{\beta^2 - \sum_t \gamma(t)^2} \tag{5.45}$$

which is the equivalent of the usual $\frac{n}{n-1}$ sample variance correction. However, we found that this correction makes little difference in practice.

The $(i,j)$th term of the numerator (5.34) can be estimated by

$$\begin{aligned}
\widetilde{\operatorname{var}}\, S_{ij} &= \frac{\sum_t \gamma(t)^2}{\beta^2} \widetilde{\operatorname{var}}\, w_{ij}(t) && (5.46)\\
&= \frac{\sum_t \gamma(t)^2}{\beta^2} . \frac{1}{\beta} \sum_t \gamma(t)(w_{ij} - S_{ij})^2 && (5.47)\\
&= \frac{\sum_t \gamma(t)^2}{\beta^2} \left[ \frac{\sum_t \gamma(t) w_{ij}^2}{\beta} - S_{ij}^2 \right] = \frac{\delta}{\beta} \eta_{ij} && (5.48)
\end{aligned}$$

where $\eta_{ij} = \frac{\sum_t \gamma(t) w_{ij}^2}{\beta} - S_{ij}^2$ and $\delta = \frac{\sum_t \gamma(t)^2}{\beta}$. The estimate of the numerator itself can be expressed in terms of these quantities by

$$\sum_{i \neq j} \widetilde{\operatorname{var}}\, S_{ij} = \frac{\delta}{\beta} \eta := \frac{\delta}{\beta} \sum_{i \neq j} \eta_{ij} \tag{5.49}$$

$\delta$ can be viewed as a correction term to allow for the increased variance when samples from nearby Gaussians “overlap” in feature space.

To estimate the $\mathbb{E} S_{ij}^2$ terms in the denominator (5.35) we follow Schäfer & Strimmer (2005) and simply replace the expectation by the squared sample values $S_{ij}^2$. As we explained in Section 5.2.2, this leads to a consistent estimator for $\hat{\alpha}$ under general asymptotics. It is possible to obtain a new estimate of $\hat{\alpha}$ at each iteration of the EM algorithm. The computation of $\hat{\alpha}$ requires two additional sets of statistics to be accumulated in the E-step, namely the sums of $w_{ij}^2$ and $\gamma^2$.

### 5.2.4 Computational issues

We briefly discuss the practical issues when estimating the shrinkage parameter from data. The estimation is computationally inexpensive since the computational cost is dominated by the computing of the $\gamma(t)$ during the E-step of the EM algorithm, which is required anyway for estimation of the other parameters. Another issue is the storage of the statistics: computing $\delta$ for each Gaussian requires the sums of $w_{ij}^2$ to be stored, which could potentially require $O(d^2)$ memory, equivalent to storing an additional covariance matrix. However, this can be avoided by the summing over $i$ and $j$ on the fly (scaling where necessary) and subtracting the $S_{ij}$ terms after all the statistics have been accumulated.

### 5.2.5 Parameter sharing

Our main references in this chapter are concerned with estimating a single full covariance matrix. When applying the techniques to GMM systems for ASR, however, there are could be hundreds thousands of covariance matrices to estimate. Without reducing the number of free covariance parameters per Gaussian, we therefore investigate the extent to which the statistics required for estimating the parameter $\alpha$ for each Gaussian may be shared across Gaussians.

Since the estimator variance reduces with the amount of training data, it is clear, from the bias-variance decomposition in Section 5.1.3 and the computations in Section 5.2.1, that the shrinkage parameter for a Gaussian will be smaller when there is more training data. It is therefore not appropriate to tie $\alpha$ across Gaussians. In order to tie parameters, it is necessary that they do not depend on the amount of training data, $\beta$.

Since they are effectively means over all samples, and because all statistics are scale-free, we might expect $\eta$ to be independent of $\beta$. We carried out an empirical analysis using the 120,000 Gaussians of our Conversational Telephone Speech (CTS) system (described in Chapter 6, page 103). Figure 5.2 shows a scatter plot of $\eta$ against $\beta$. The Pearson correlation coefficient between the two variables is -0.053, which is a small, but statistically significant correlation. Although there may be a better approach, we propose to tie $\eta$ across Gaussians.

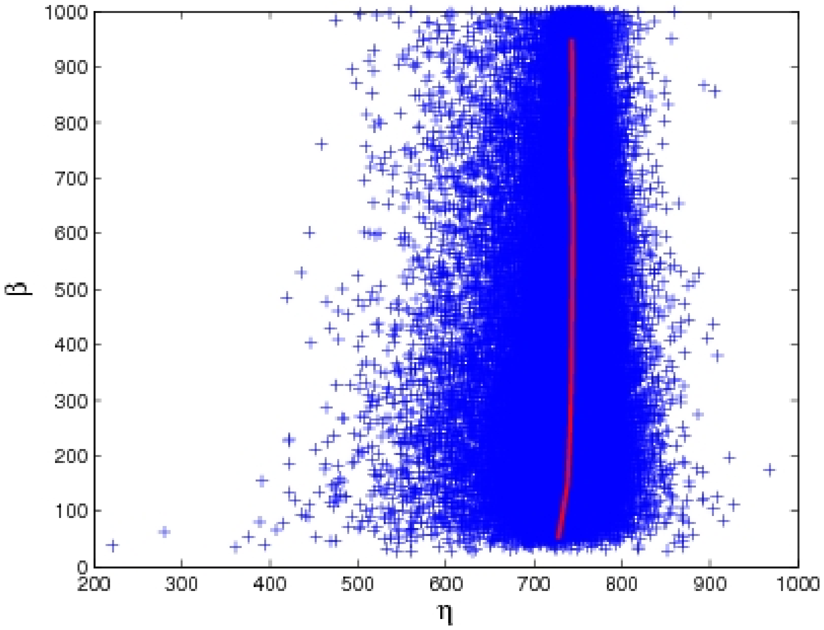

Figure 5.2: Scatter plot showing values of $\eta$ and $\beta$ for 120,000 Gaussians, and a mean trendline.

Pooling the denominator estimates of $\mathbb{E}S_{ij}^2$ is less straightforward. Decomposing

$$\mathbb{E}S_{ij}^2 = (\mathbb{E}S_{ij})^2 + \operatorname{var} S_{ij} \tag{5.50}$$

we see that the expression consists of a expectation term which we would expect to be constant with $\beta$ and a variance which reduces with $\frac{1}{\beta}$. This is illustrated by our empirical analysis of these statistics for the CTS Gaussians, shown in Figure 5.3 $S_{ij}$ has a Wishart distribution, and so the total variance can be approximated by

$$\sum_{i \neq j} \operatorname{var} S_{ij} \approx \frac{2\delta\eta}{\beta} \tag{5.51}$$

so that

$$\mathbb{E}S_{ij}^2 \approx (\mathbb{E}S_{ij})^2 + \frac{2\delta\eta}{\beta} \tag{5.52}$$

This is illustrated in Figure 5.4, plotting the sums $S_{ij}^2$ against $\delta/\beta$ for each Gaussian. We would expect the bias term to be constant, and we can obtain a shared value for it by averaging (5.35) by

$$C = \sum_{i \neq j} S_{ij}^2 - \frac{2\delta\bar{\eta}}{\beta} \tag{5.53}$$

across all Gaussians. In Figure 5.4, we indeed find the equation of the trendline to be given by $C + \frac{2\delta\eta}{\beta}$ with a good measure of fit. For the denominator, we can then use

$$\mathbb{E}S_{ij}^2 = \bar{C} + \frac{2\delta\bar{\eta}}{\beta} \tag{5.54}$$

### 5.2.6 Comparisons with the Bayesian approach

Recall the Bayesian formulation (Equation 5.22):

$$U = (1-\alpha)S + \alpha D \tag{5.55}$$

$$= \frac{S}{\beta+\tau} + \frac{\tau D}{\beta+\tau} \tag{5.56}$$

When the shrinkage statistics are shared across all Gaussians in the system, is the same form of smoothing recovered? Using the shared statistics above, the

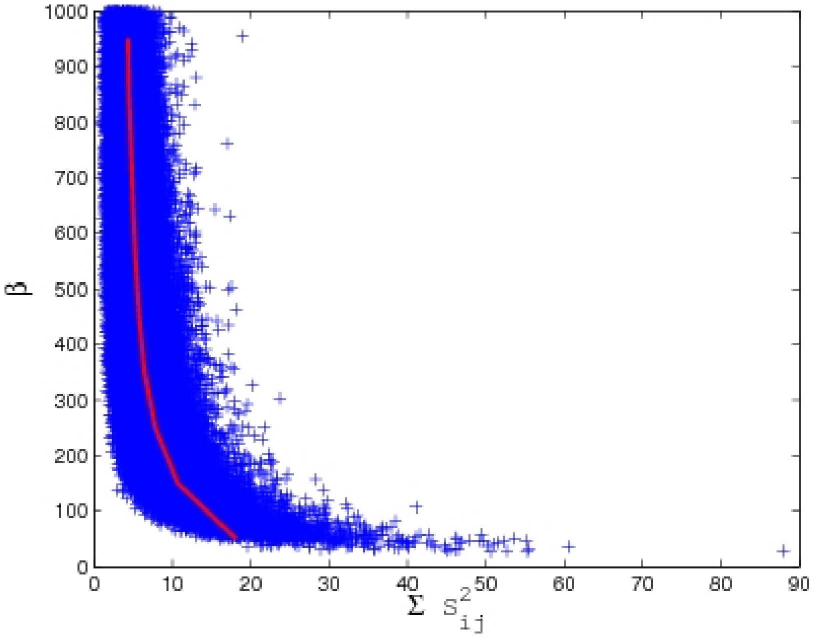

Figure 5.3: Scatter plot showing values of $\sum_{i \neq j} S_{ij}^2$ and $\beta$ for 120,000 Gaussians, and a mean trendline.

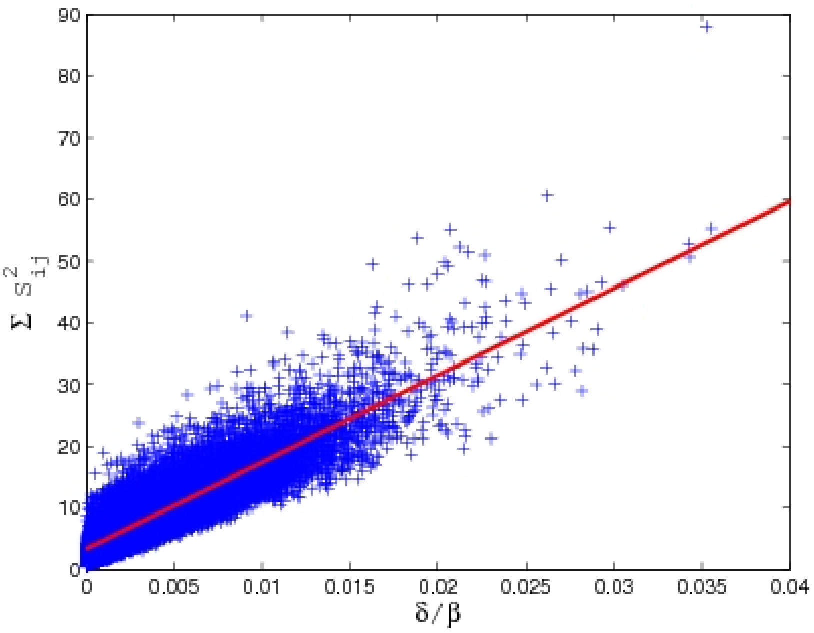

Figure 5.4: Scatter plot showing values of $\sum_{i \neq j} S_{ij}^2$ and $\frac{\delta}{\beta}$ for 120,000 Gaussians, and a mean trendline with equation $y = 3.3 + 1410x$. The $R$-squared coefficient measuring the model fit is 0.77.

shrinkage parameter is given by

$$\alpha = \frac{\eta\delta/\beta}{C + 2\eta\delta\beta} \tag{5.57}$$

$$= \frac{\eta\delta/C}{\beta + 2\eta\delta/C} \tag{5.58}$$

Comparing to (5.56) we see that this is similar to using a prior with weight $\eta\delta/C$, except the weighting is doubled in the denominator, so that in the limit as the quantity of training data is reduced towards zero, the off-diagonal elements are reduced by half, rather than vanishing to zero. This is equivalent to using, as prior,

$$p(\Sigma) = \mathcal{W}^{-1}(2\tau + d + 1, 2\tau.\frac{1}{2}(D + S)) \tag{5.59}$$

for which

$$U = \frac{S}{\beta + 2\tau} + \frac{\tau(S + D)}{\beta + 2\tau} \tag{5.60}$$

The difference can be explained by the fact the original Bayesian approach models the diagonal prior as fixed, whilst the shrinkage formulation takes into the variance of the diagonal elements into account.

## 5.3 Summary

In this chapter we introduced the shrinkage estimator, an alternative estimator for the full covariance matrix that corrects for the fact that the standard sample covariance matrix is not consistent under weaker assumptions about the asymptotic nature of the dimensionality of the feature space relative to the number of training samples.

We explained how the optimal shrinkage parameter for these models may be estimated consistently under the weaker asymptotic assumptions, and obtained formulae for estimating the parameter within a GMM system. Based on investigations using a large-vocabulary ASR system, we suggested a method for tying the parameter across Gaussians. We compared the shrinkage technique with a Bayesian approach.

# Chapter 6

# Covariance modelling experiments

## 6.1 TIMIT phone recognition

### 6.1.1 The task

We carried out all early covariance modelling experiments on the TIMIT corpus. The corpus consists of 6300 read sentences: 630 speakers, drawn from eight dialect regions of the US, spoke ten sentences each. Two single sentences (the 'SA' sentences) were repeated by all speakers. To avoid distorting the results, we ignored these sentences throughout training and testing, leaving 5040 sentences. The remaining sentences were designed to provide good phonetic coverage (the 'SX' sentences) and phonetic diversity (the 'SI' sentences). All sentences were recorded in clean conditions. Phonetic transcriptions are provided for all recordings using a 61-phone set.

We built a system for the standard phone recognition task using the TIMIT corpus. The standard training set consists of 3696 sentences, and we used the core test set, 192 sentences from 24 speakers. Following standard practice, performance was evaluated using a reduced 39-phone set (Lee & Hon, 1988), described in Table B.9 in Appendix B.2 . The task has several advantages for developing new techniques: the relatively small size of the corpus, and the fact that only phone-level decoding is required, makes it quick to train and test new models; and the recording conditions and speaking style reduce the need for context-dependent modelling and noise reduction techniques

Since our particular interest lies in the case when the amount of training data is small, we conducted experiments where the amount of training data is artificially reduced. Utterances were removed at random from the training corpus. In the smallest case, data consisted of just 10% of the full training set.

### 6.1.2 Baseline system design

Our baseline system was a monophone HMM system using 48 phone models, each with three emitting states. The models were trained using the phonetic transcriptions provided, with the 61-phone set collapsed to the 48 phones, listed in Table B.9. The acoustic feature vector consisted of 12 MFCCs plus energy component, their deltas and double-deltas. A phone-based bigram language model was used for decoding. Following an initial search, the language model scaling factor and insertion penalty were fixed for all experiments at 5.0 and 2.5 respectively. When computing phone accuracy scores, the silence phone, 'sil', was ignored. Including it, which is the normal practice, increases the accuracy by more than 3%.

The number of Gaussians per phone state was increased using the mixing up procedure described in Section 2.3.3. Using the full training set, we obtained accuracy results on the test set with the number of Gaussians increasing from 1 to 100. These are shown in Figure 6.1 and Table B.1 (most of the large tables are contained in Appendix B) . The results indicate a performance peak for the diagonal covariance models of 66.0% accuracy, attained at 72 Gaussians per state. The figures are consistent with those obtained by others using a similar system, for example, Valtchev *et al.* (1997). The largest system shown in the table, with 100 Gaussians, contained 7800 mean and covariance parameters per state. For comparison, a full covariance system with 12 Gaussians has 9828 parameters.

The same mixing up procedure was carried out to train GMMs on each of the reduced training sets. Phone accuracy results with these reduced-data models are compared Figure 6.2 and Table B.2 and. It can be seen that when the models have few parameters, the size of the training set makes little difference

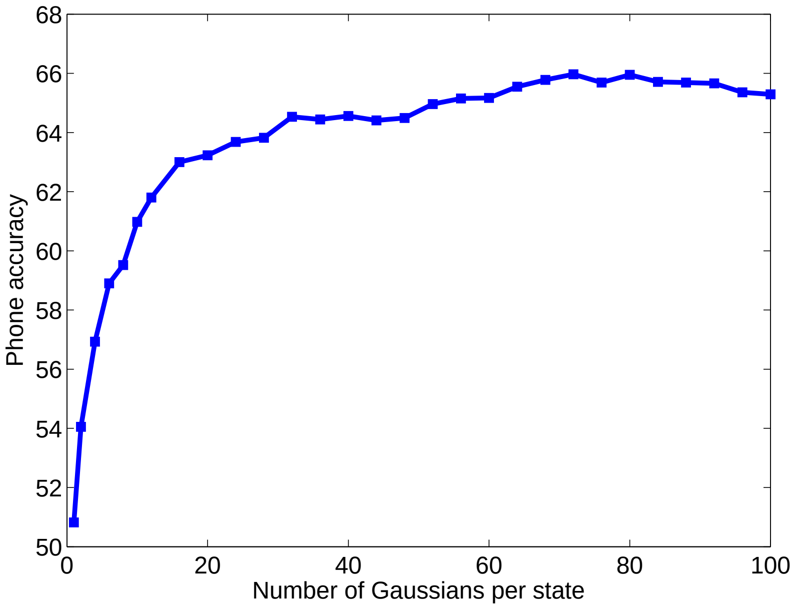

Figure 6.1: Phone accuracy of diagonal covariance models trained on the full training set. See Table B.1 for data.

to performance; as the number of parameters increases, more data is required to achieve good performance.

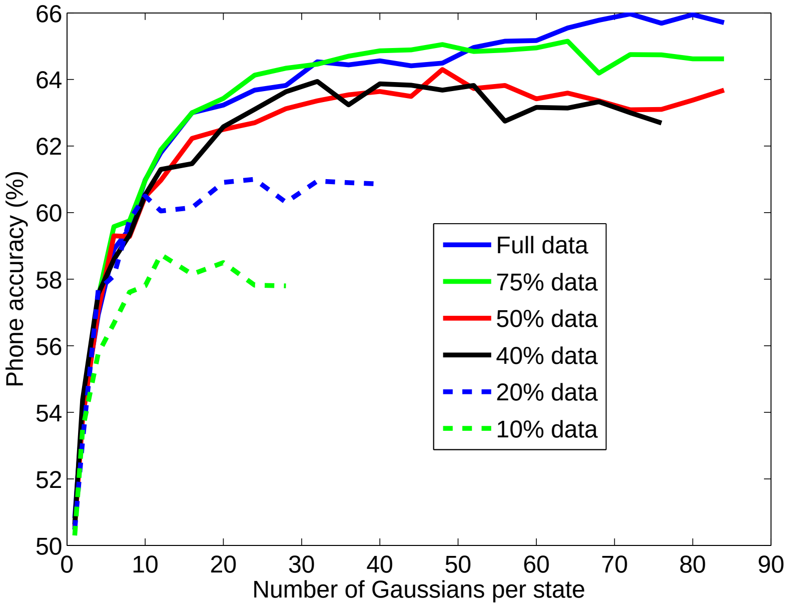

Figure 6.2: Phone accuracy of diagonal covariance models trained on selected subsets of the full training set. See Table B.2 for the data.

In each case, full covariance GMMs were initialised from diagonal covariance models with the same number of Gaussians. These models were used to accumulate full covariance statistics centred on the existing means. An alternative scheme that has been suggested is to carry out the mixing up procedure with full covariance models (this is discussed for semi-tied covariance matrices in Gales, 1999). The suggested advantage to this method is that it avoids Gaussians redundantly modelling feature correlations, allowing a fixed number of Gaussians to instead model the multimodal nature of the distribution more effectively. However, we did not do this: we found that poorly-performing covariance modelling techniques cause rapid over-training when the number of Gaussians is still small, making the mixing up ineffective, and providing poor comparisons between different techniques when the number of Gaussians is large. Additionally, the former technique keeps the computational cost of training much lower by avoiding the need for many Baum-Welch re-estimations with full covariance models.

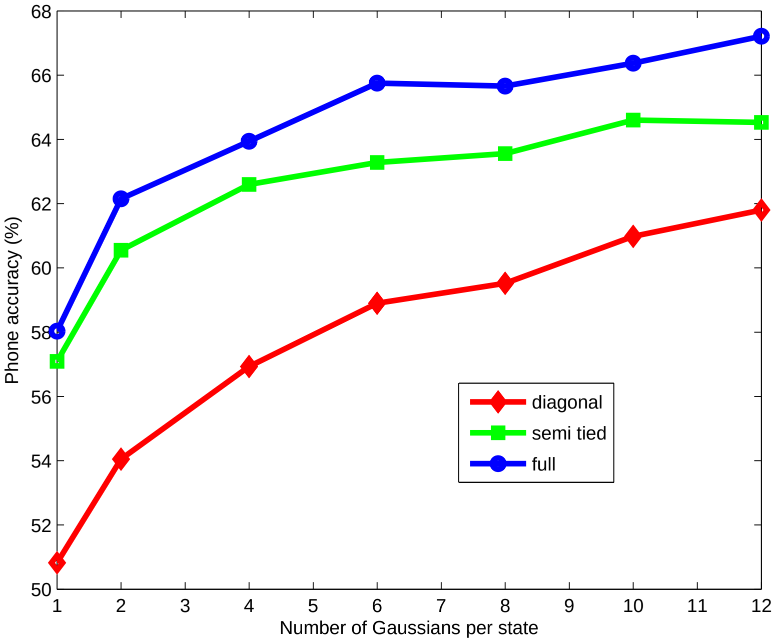

Figure 6.3: Phone accuracy of covariance models trained on the full training set, with varying number of Gaussians. See Table B.5 for the data.

Figure 6.3 compares phone accuracy results with varying number of Gaussians for three standard covariance models of varying complexity: diagonal covariance, semi-tied covariance, and full covariance. The figures are contained in Table B.5. As discussed above, the semi-tied and full covariance models were initialised directly from the diagonal Gaussians after mixing up. In the semi-tied covariance case, the transforms were tied at the state level. Note that the semi-tied system achieves slightly lower performance compared to the full covariance system when there is a single Gaussian per state, when they would be expected to perform the same – this because the number of iterations used to re-estimate the transforms was limited. When training baseline full covariance systems, we use a naive technique suggested by Povey (2009): each covariance matrix is estimated by the sample covariance matrix, unless there are fewer than $d$ (in this case, 39) samples, in which case, we back off to the diagonal matrix. We refer to this as "naive" full covariance.

In these experiments we are more interested in comparing covariance modelling techniques with varying hyper-parameters and varying quantities of data, rather than exhaustively optimising every parameter for every condition. We assume correct the findings of Axelrod *et al.* (2005), discussed in Section 3.4, that full covariance models can achieve higher performance than other models, regardless of the number of Gaussians. Therefore we do not specifically seek to demonstrate that our full covariance models are always capable of outperforming semi tied models and diagonal models; this would require careful optimisation of the latter systems. Rather, results with these models are shown for comparison: they are not necessarily indicative of the best performance attainable.

In presenting results here, some choices must be made. When showing the effects of varying covariance hyperparameters, we generally fix the number of Gaussians at 12 per state, and illustrate the effects with models trained using three different quantities of training data: on 10%, 50%, and the full training set. When no hyperparameters are involved, as with the naive full covariance estimator and the shrinkage estimator, we show performance changes with a greater range of sizes of training data. We also show the how these models perform when the number of Gaussians is varied, again using the three training sets.

### 6.1.3 Gaussian graphical modelling

We conducted experiments with sparse Gaussian graphical models, which we trained using the $l_1$-penalised maximum likelihood method described in Section 4.4. Figures 6.4, 6.5 and 6.6 show phone accuracy results for three different training data conditions when the penalty parameter, $\rho$ is increased from zero (which corresponds to the naive full covariance system). All models have 12 Gaussians per state. The results are contained in Table B.3 We counted the number of non-zero parameters in the resulting GM systems; these are included in the table.

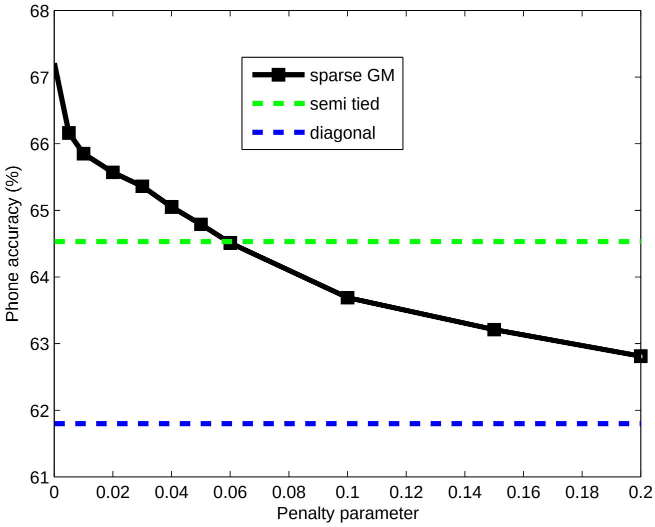

Figure 6.4: Phone accuracy of sparse GM models with varying penalty parameter, $\rho$, trained on the full training set.

In the two cases where the full covariance models have higher performance than the diagonal models, the steady decline in performance as the penalty parameter is increased is consistent with the results reported in (Bilmes, 2000b). In the 10% data case the performance is dramatically improved by penalising the likelihood, illustrating the benefits of covariance regularisation; however, the performance attained remains lower than the diagonal models. Given the more positive results attained using the shrinkage estimator, described in the follow-

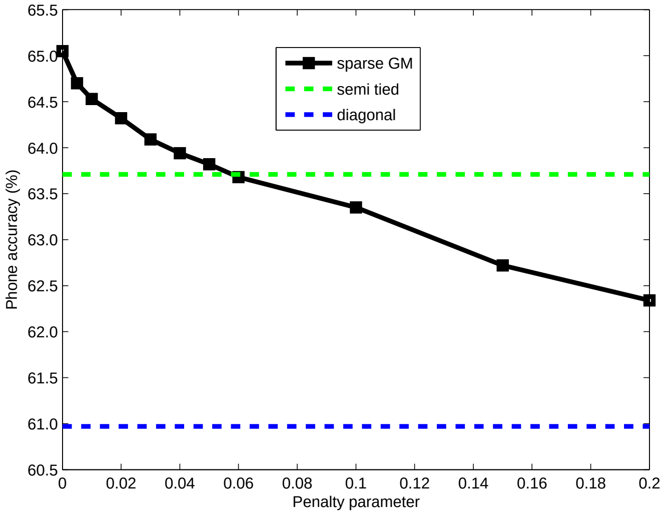

Figure 6.5: Phone accuracy of sparse GM models with varying penalty parameter, $\rho$, trained on 50% of the training set.

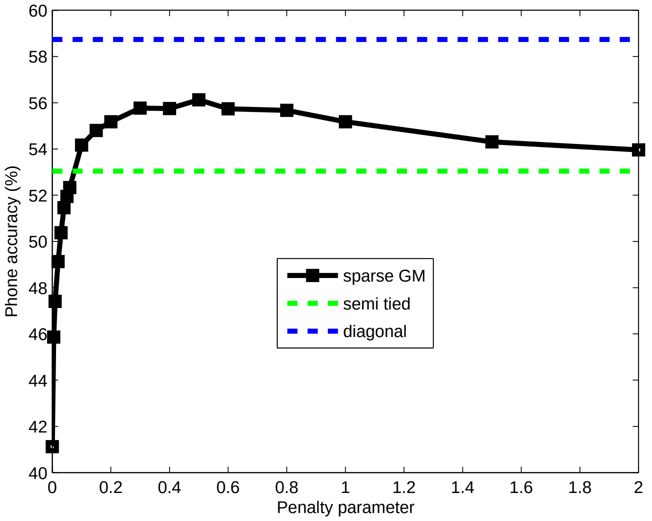

Figure 6.6: Phone accuracy of sparse GM models with varying penalty parameter, $\rho$, trained on 10% of the training set.

ing section, we did not conduct further detailed experiments using the Gaussian graphical modelling techniques.

### 6.1.4 Shrinkage estimator

In this section we present results using the shrinkage estimator as an alternative to the naive full covariance estimator. We investigated the performance of the models with varying quantities of training data, and varying number of Gaussians. Note that the term *shrinkage estimator* could refer to any models with off-diagonal smoothing. However, here we use it more specifically to refer to models where the shrinkage parameter $\alpha$ was analytically obtained, as described in Section 5.2. We compared the performance of these models with those where the prior weight, $\tau$, was chosen heuristically.

#### Varying prior weight

Figures 6.7, 6.8 and 6.9 compare the performance of smoothed models where the shrinkage parameter is estimated directly from the training data to models where a prior weight is chosen heuristically. All models have 12 Gaussians per state. Phone accuracy results are shown with three different training sets, for a variety of selections of prior weights. The result are contained in Table 6.1. Although technically there is a slight difference, we use a prior weight of zero to denote a naive full covariance system.

These results appear to validate the method used to determine the optimal shrinkage parameter: in all the data conditions investigated, the shrinkage estimator performance never falls more than 0.6% below the best-performing smoothed model, and the gap is generally less than that (in four of the cases, the difference is no worse than 0.1%). There is no clear correlation between the optimal parameter $\tau$ and the quantity of training data, which supports the conclusions of Section 5.2.6.

#### Varying training data

Figure 6.10 compares the performance of the shrinkage estimator with the naive full covariance estimator – and also semi-tied and diagonal systems – when the amount of training data is varied. The results are contained in Table 6.1. It can

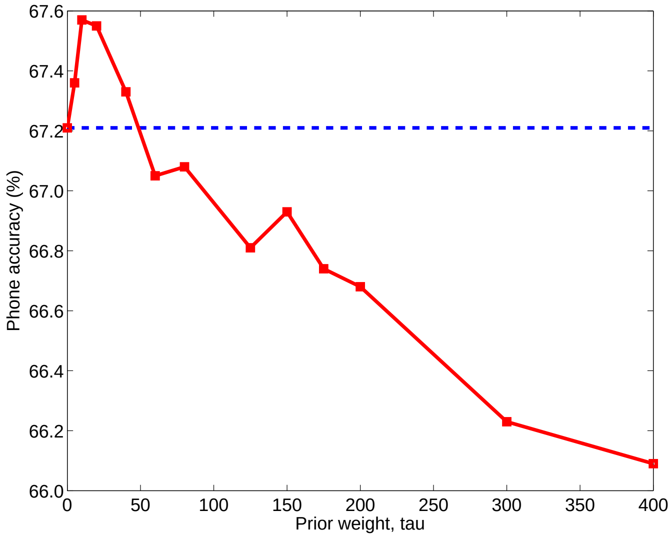

Figure 6.7: Phone accuracy of smoothed full covariance models, with varying prior $\tau$ (solid red) compared with analytic shrinkage parameter (dashed blue), trained on the full training set.

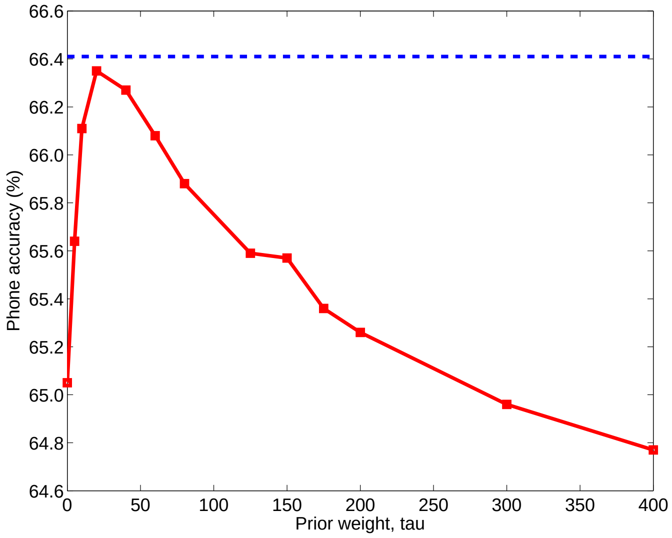

Figure 6.8: Phone accuracy of smoothed full covariance models, with varying prior $\tau$ (solid red) compared with analytic shrinkage parameter (dashed blue), trained on 50% of the training set.

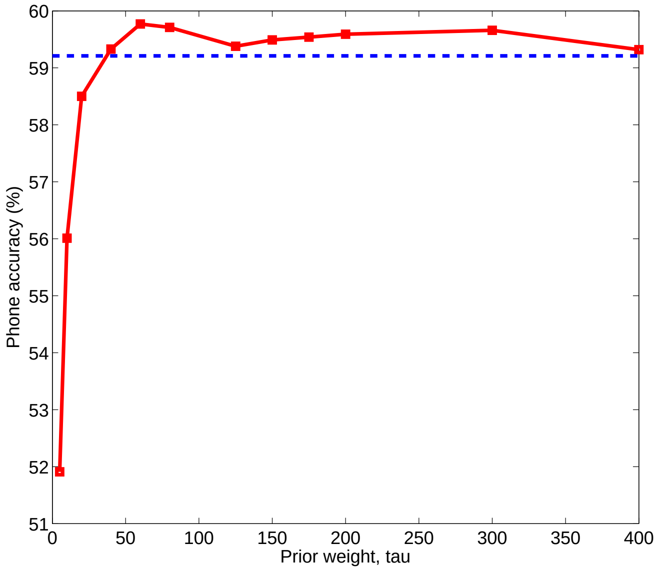

Figure 6.9: Phone accuracy of smoothed full covariance models, with varying prior $\tau$ (solid red) compared with analytic shrinkage parameter (dashed blue), trained on 10% of the training set.

| | Proportion of full training set used | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Prior $\tau$ | 10% | 20% | 30% | 40% | 50% | 60% | 75% | 100% |
| 0 | 41.1 | 56.4 | 61.8 | 63.5 | 65.1 | 65.6 | 66.6 | 67.2 |
| 5 | 51.9 | 60.5 | 63.6 | 64.2 | 65.6 | 66.3 | 66.8 | 67.4 |
| 10 | 56.0 | 62.0 | 64.0 | 64.6 | 66.1 | **66.6** | 67.1 | **67.6** |
| 20 | 58.5 | 62.7 | **64.4** | 64.8 | **66.4** | 66.5 | 67.1 | **67.6** |
| 40 | 59.3 | **63.2** | 64.3 | 65.0 | 66.3 | **66.6** | **67.2** | 67.3 |
| 60 | **59.8** | 63.1 | 64.3 | **65.1** | 66.1 | 66.3 | 67.1 | 67.1 |
| 80 | 59.7 | 62.9 | 63.9 | 64.9 | 65.9 | 66.5 | 67.1 | 67.1 |
| 100 | 59.5 | 62.7 | 64.2 | 64.7 | 65.8 | 66.3 | 66.9 | 67.0 |
| 125 | 59.4 | 62.6 | 63.8 | 64.6 | 65.6 | 66.3 | 66.5 | 66.8 |
| 150 | 59.5 | 62.6 | 63.9 | 64.5 | 65.6 | 66.3 | 66.3 | 66.9 |
| 175 | 59.5 | 62.3 | 63.6 | 64.3 | 65.4 | 66.3 | 66.5 | 66.7 |
| 200 | 59.6 | 62.3 | 63.5 | 64.2 | 65.3 | 66.0 | 66.5 | 66.7 |
| 300 | 59.7 | 62.3 | 63.3 | 63.9 | 65.0 | 65.7 | 66.1 | 66.2 |
| 400 | 59.3 | 62.4 | 63.2 | 63.9 | 64.8 | 65.4 | 65.9 | 66.1 |
| Diagonal | 58.7 | 60.1 | 61.7 | 61.3 | 61.0 | 61.6 | 61.9 | 61.8 |
| Shrinkage | 59.2 | 63.3 | 64.2 | 65.0 | 66.4 | 66.4 | 67.1 | 67.2 |
| Semi tied | 53.0 | 60.2 | 62.5 | 63.3 | 63.7 | 64.2 | 64.2 | 64.5 |

Table 6.1: Phone accuracy of with varying prior $\tau$, 12 Gaussians per state.

be seen that the system using the shrinkage estimator outperforms all the other systems, for all quantities of training data. The performance of the standard full covariance system drops rapidly as the amount of training data is reduced, whilst the shrinkage system maintains its robustness. At 10% data, it continues to outperform the diagonal system.

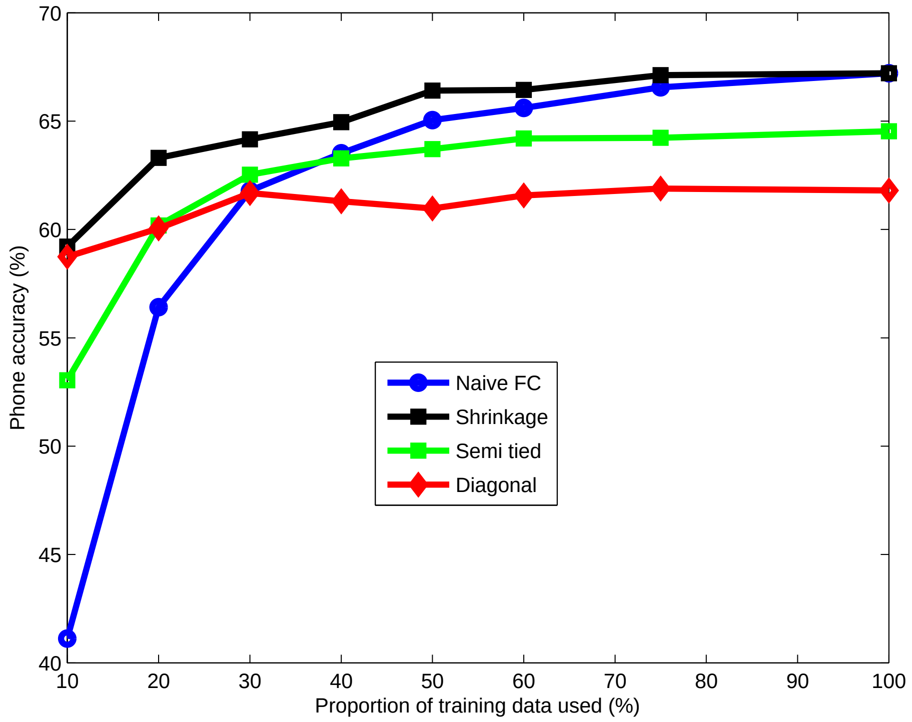

Figure 6.10: Phone accuracy of covariance models with 12 Gaussians per state, with varying amounts of training data. See Table 6.1 for the data.

Figure 6.11 is a similar plot, but shows only smoothed full covariance systems: the shrinkage estimator, and four systems with the prior weights set to 25, 50, 100 and 150 respectively (additional data is contained in Table B.4). This illustrates the effect of varying the prior weight, and supports the conclusion that the optimal prior weight does not vary with the quantity of training data. It also demonstrates that the shrinkage estimator achieves close to the best possible performance over all the data conditions. However, the results demonstrate that it is possible for a single appropriately chosen prior weight – 50 in this case – to achieve equally good, or better, results across almost all conditions.

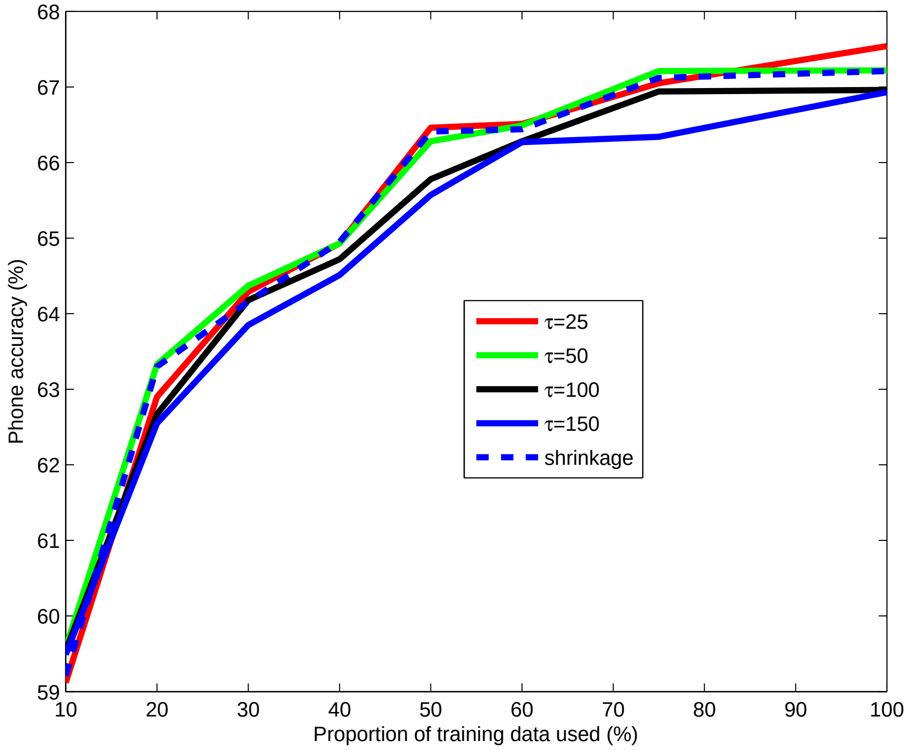

Figure 6.11: Phone accuracy of smoothed full covariance models, with varying amounts of training data. Data are contained in Table 6.1 and additionally Table B.4

### Varying number of Gaussians

For completeness, we also investigated the performance of the various covariance models with varying numbers of Gaussians per state. Increasing the number of Gaussians increases the modelling power, but it effectively reduces the amount of training data available to estimate the covariance parameters of each Gaussian, causing problems for models which do not generalise well. Figures 6.12, 6.13 and 6.14 show phone accuracy results for the three training data sets. The data are contained in Tables B.5, B.6 and B.7. Table B.8 gives the total number of mean and variance parameters in each system.

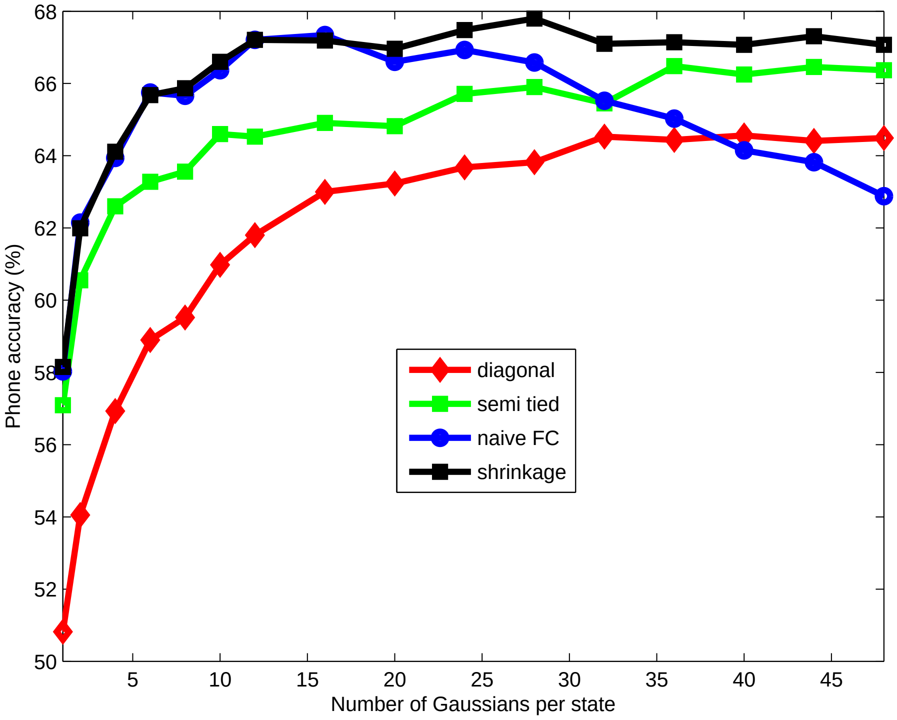

Figure 6.12: Phone accuracy of covariance models trained on the full training set, with varying number of Gaussians. See Table B.5 for the data.

The results show that, initially, increasing the number of Gaussians in the system increases performance for all covariance models. When the number of Gaussians is small, the naive full covariance system outperforms models with fewer covariance parameters; however, when the number of Gaussians increases, the performance of the naive models begins to decline first – and most rapidly – due to the failure of these models to generalise. The effect is most pronounced, and begins to occur at a lower number of Gaussians, when the amount of training data is smaller. In contrast, the shrinkage systems, despite having the same number of parameters, consistently achieve the highest performance.

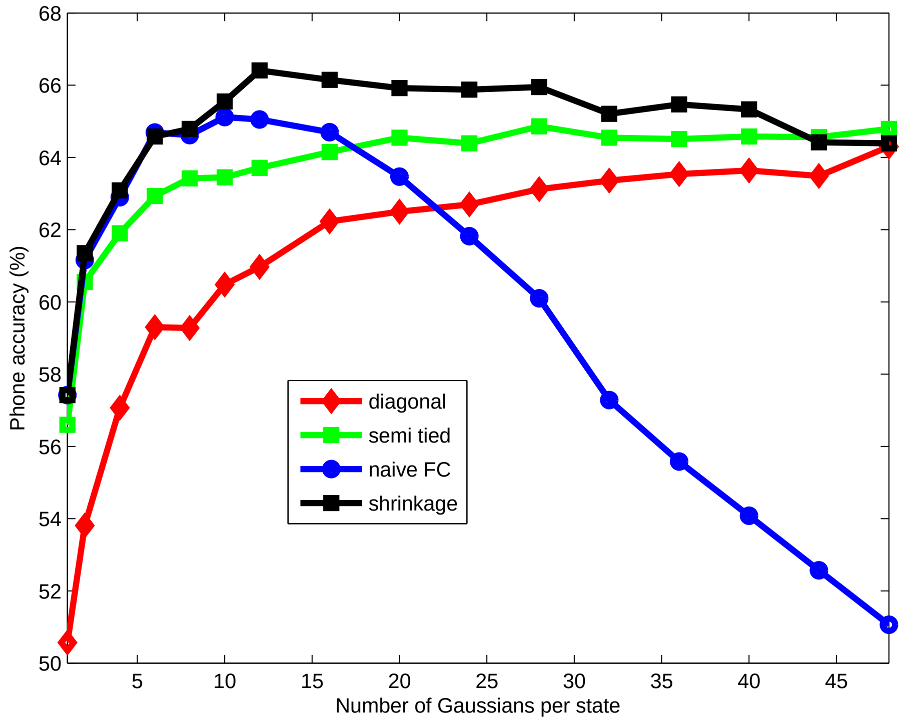

Figure 6.13: Phone accuracy of covariance models trained on 50% of the training set, with varying number of Gaussians. See Table B.6 for the data.

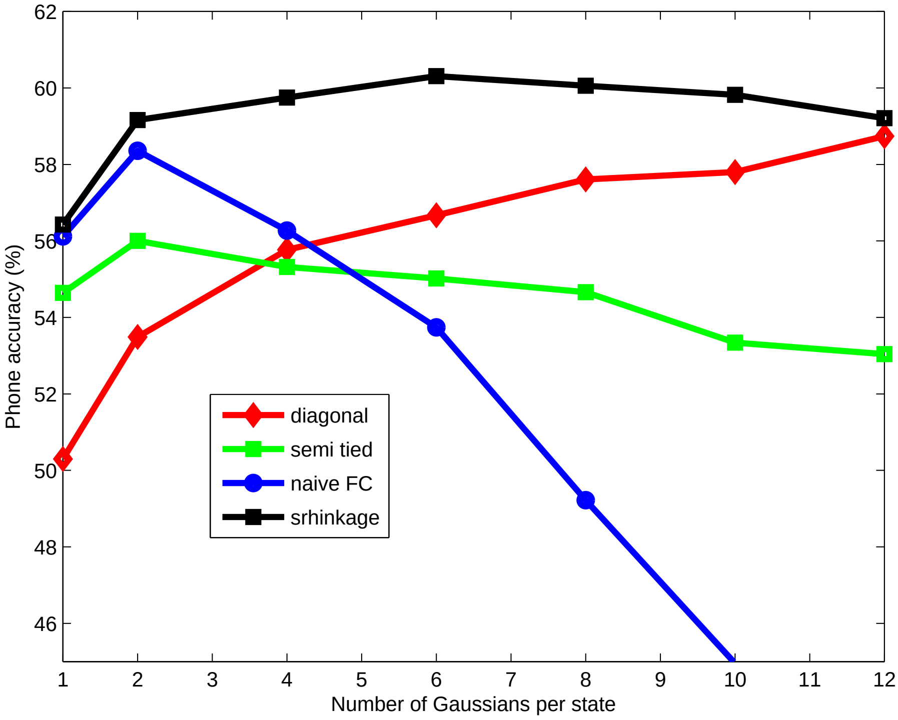

Figure 6.14: Phone accuracy of covariance models trained on 10% of the full training set, with varying number of Gaussians. See Table B.7 for the data.

### Summary

Under all the quantities of training data used for model training, the highest phone accuracy results on the test set were achieved by full covariance models smoothed with a diagonal prior. The shrinkage estimator, with analytically obtained shrinkage parameter, achieved close to the optimum in all conditions investigated. However, it was usually possible to achieve a higher score with a constant well-chosen prior weight across all Gaussians, and a single weight, $\tau = 50$ performed at least as well as the shrinkage estimator in almost all data conditions.

As would be expected, the improvement derived from smoothing, compared to the naive full covariance estimation, was greatest when the quantity of training data was smallest. As a similar effect was observed when the number of Gaussians was increased: this effectively reduced the quantity of training data available to train the parameters of each Gaussian, and again, the beneficial effect of the smoothing was greater. We did not exhaustively search for the optimal diagonal or semi-tied covariance systems, but the results obtained do seem to support the findings of Axelrod *et al.* (2005), that full covariance models are capable of performance not achievable simply by increasing the number of Gaussians. For example, in the full data condition, the peak performance of the shrinkage estimator exceeds the peak diagonal performance by 1.2%, including cases when the diagonal system has more parameters in total. In the 50% and 10% data conditions the shrinkage estimator achieved the highest peak performance over 1-48 Gaussians per state. However, we cannot make definitive claims about the peak performance attainable by the diagonal and semi-tied systems.

## 6.2 Conversational telephone speech recognition

### 6.2.1 Experimental setup

We performed large-vocabulary speech recognition experiments on a conversational telephone speech (CTS) task. Our test set was the the NIST Hub5 2001 evaluation set[1], comprising 6 hours of conversational telephone speech from the

[1] http://www.nist.gov

Switchboard-1, Switchboard-2 and Switchboard-cellular corpora, with 60 male and 60 female speakers. The reference transcriptions contained a total of 62,890 words.

Our baseline system was derived from the 2005 AMI recogniser (Hain *et al.*, 2005a,b,c) which was evaluated on the same CTS task. From this system we used:

- Bigram and trigram language models interpolated from a variety of corpora – including Switchboard, Call Home and Fisher – smoothed using Kneser-Ney discounting (Ney *et al.*, 1994).

- A pronunciation lexicon derived from the accent-independent Unisyn lexicon (Fitt, 2000), with manual corrections for out-of-vocabulary pronunciations.

- Automatic segmentations of the test data into individual utterances using statistical speech activity detection.

- Baseline cross-word triphone acoustic models, with standard three-state topology, clustered with a phonetic decision tree. These models comprised approximately 120,000 diagonal-covariance Gaussians – 16 per state – and were trained using maximum likelihood estimation. The acoustic feature vector contained 12 PLP plus energy coefficients, their delta and double deltas, with CMN and CVN applied on an entire-channel basis.

These features of the system are described in more detail in (Garau, 2008). For our acoustic model training, we used the same training set as that used to train the baseline models, 277 hours of speech from the Switchboard-1, Switchboard-2 and Call Home corpora. Our models were initialised from the diagonal-covariance models and we did not alter the number of Gaussians, or the triphone clustering.

We used HTK's HDecode[1] tool with a bigram language model to generate lattices for the segmented test utterances, using the baseline acoustic models. To reduce the computational cost of decoding with the full covariance models, we instead used acoustic rescoring of these baseline bigram lattices. For consistency with the full covariance results, all the results with diagonal systems we present

[1] http://htk.eng.cam.ac.uk/

here used the same rescoring technique; we found that this typically resulted in a word error rate (WER) around 0.2% higher than when a full decoding was used. The final lattices were then rescored with a trigram language model to produce a one-best transcription. We used the NIST scoring tools to obtain the WER. In all cases, language model probabilities were scaled by 12.0.

### Statistical significance

We used the NIST scoring tools to check the statistical significance of WER differences between key systems. Significance was evaluated using a matched-pair sentence segment word error (MAPSSWE) test (Pallett *et al.*, 1990). Unless stated otherwise, all WER differences described in the following sections as statistically significant were significant at the $p < 0.1\%$ level; in all cases a two-tailed test was used.

It would be cumbersome to quote statistical significance for all possible pairs of results. However, a simple rule of thumb (due to Povey, 2003) may be used to approximately gauge the significance of those results for which significance is not explicitly stated. Suppose that there are $n$ tokens in the reference transcription, and in the transcription produced by the recogniser, each of these has probability $p$ of being incorrectly expressed. Assuming errors are independent, the distribution of total errors, $E$ is then $Bin(n, p)$, which for large $n$ can be accurately approximated by a normal distribution with the same mean and variance, $\mathcal{N}(np, np(1-p))$. The proportion of errors $E/n$, equivalent to the WER has distribution $\mathcal{N}(p, \frac{1}{n}p(1-p))$. $p$ can be estimated by the observed WER. Assuming a constant variance, a change in error rate is significant at the 5% level if it exceeds 2 standard deviations, and at the 1% level if it exceeds 2.6 standard deviations.

For the CTS task, there are 62,890 tokens. For error rates close to 30%, we estimate the standard deviation by 0.18%. Therefore, using this approximate approach, we can judge a change in WER between two systems to be significant at the 5% level if it exceeds 0.4%, and at the 1% level if it exceeds 0.5%. In practice this approach is often found to be conservative: we found some pairs of results differing by only 0.2% absolute WER to be significantly different at the 0.1% level using the MAPSSWE test with the NIST scoring tools.

### 6.2.2 Diagonal covariance system refinements

We implemented the acoustic modelling refinements described in Section 2.4. The feature vector was extended to include third-differential coefficients, and a global HLDA projection was applied to reduce the dimensionality from 52 back to 39. Additionally, we applied speaker adaptation with block-diagonal CMLLR transforms with 32 regression classes per speaker. (A single transform was used for silence models).

Finally, we performed SAT, estimating CMLLR transforms for each training speaker using the same regression classes. The models were re-trained using these transforms. We repeated this procedure for two iterations. The SAT models were, of course, always used with CMLLR on the test speakers. We did not apply any vocal tract length normalisation (VTLN), though we would expect it to give further improvements on the results shown here. Table 6.2 shows the results from the diagonal covariance systems. WER improvements down the table are all statistically significant.

| System | Bigram LM | Trigram LM |
|---|---|---|
| Baseline | 40.3 | 37.2 |
| HLDA | 38.5 | 35.5 |
| CMLLR | 36.5 | 34.5 |
| HLDA + CMLLR | 35.6 | 33.3 |
| HLDA + SAT + CMLLR | 35.3 | 32.9 |

Table 6.2: %WER results for diagonal covariance system refinements with bigram and trigram language models.

### 6.2.3 Full covariance systems

As in the TIMIT experiments, we compared the full covariance models to semi-tied covariance matrices. We trained full-parameter semi-tied transforms after the application of the global HLDA transform. The transforms were estimated just once; the other Gaussian parameters were updated with a further two reestimations. We investigated transforms tied at monophone level and also at the monophone state level.

| System | Trigram WER |
|---|---|
| Baseline | 37.2 |
| STC (monophone) | 36.0 |
| STC (monophone state) | 35.6 |
| CMLLR | 34.5 |
| CMLLR + STC (monophone) | 34.1 |
| CMLLR + STC (monophone state) | 34.3 |
| HLDA | 35.5 |
| HLDA + STC (monophone) | 35.1 |
| HLDA + STC (monophone state) | 34.8 |
| HLDA + CMLLR | 33.3 |
| HLDA + CMLLR + STC (monophone) | 33.2 |
| HLDA + CMLLR + STC (monophone state) | 33.1 |
| HLDA + SAT + CMLLR | 32.9 |
| HLDA + SAT + CMLLR + STC (monophone) | 33.3 |
| HLDA + SAT + CMLLR + STC (monophone state) | 33.4 |

Table 6.3: %WER results for STC systems.

We again applied speaker adaption. For adaptation of a full-covariance system, CMLLR has the advantage that it can be formulated as a feature-space transform rather than a model-space transform, so it is not necessary to recompute full covariance matrices. For the adaptation of both the full covariance and semi-tied systems, we implemented an approximation described by Povey & Saon (2006) and Sim & Gales (2005): the transforms are obtained by optimising the objective function given in Equation 2.55 (page 29), using just the diagonal elements of the covariance matrix. In fact the results for the full covariance systems shown below use an even simpler approach – we just use the CMLLR transforms estimated for the diagonal models used for initialisation ("initial CMLLR"), so that the same transforms were used with all full covariance models; the same method is employed in the experiments in Chapter 8. Limited investigations found the approximate diagonal method to reduce the WER by a further 0.1%.

Results from various semi-tied systems are shown in Table 6.3. We found that STC is ineffective when used with the CMLLR transforms (improvements when STC was used were not statistically significant) – this is because the phone-specific semi-tied transforms are effectively absorbed into the CMLLR transforms. Properly adapting these systems requires a more sophisticated approach (Gales, 1997).

Following the approach taken in the TIMIT experiments, we initialised the full covariance models directly from the final set of diagonal-covariance Gaussians. We found that the estimation of the full-covariance models was quick to converge, so the models used for the results presented below were ML-trained using just one iteration with full covariance, keeping the Gaussian means fixed. We found that a further mean and variance re-estimation reduced the WER by 0.1%. Using just one iteration allowed multiple prior weights to be investigated rapidly, since the full covariance E-step statistics could be re-used for every smoothed model.

Table 6.4 shows the refinements applied to the shrinkage system, where the shrinkage parameter was estimated analytically. Firstly, models were initialised from the diagonal HLDA models, and the CMLLR transforms from these models applied. We then investigated the effect of pooling the shrinkage statistics $\eta$ and $C$, as described in Section 5.2.5). This resulted in a 0.2% absolute WER improvement. (For interest, we found $\eta = 740$, $C = 3.1$ and a mean $\delta = 0.75$, giving an average shrinkage parameter $\alpha = 0.23$, equivalent to an average smoothing parameter $\tau = 118$). All these improvements were statistically significant. The same result was obtained when pooling globally and at monophone level. We then applied the same techniques using SAT: here, the full covariance model was initialised from the diagonal HLDA+SAT model, with the previous CMLLR transforms for both training and test speakers used. The equivalent diagonal models are shown for comparison. It can be seen that the shrinkage full covariance models result in a substantial improvement over the diagonal models, 2.7% and 2.5% absolute WER for the non-SAT and SAT models respectively.

| System | Trigram WER |
|---|---|
| HLDA + Shrinkage | 32.2 |
| + initial CMLLR | 30.8 |
| + pooled $\eta$ and $C$ | 30.6 |
| + SAT | 30.4 |
| HLDA + CMLLR | 33.3 |
| HLDA + SAT + CMLLR | 32.9 |

Table 6.4: %WER results for shrinkage systems with additional refinements.

We again compared models where the shrinkage parameter was estimated directly from the training data to models using a manually specified prior weight. For non-SAT systems, the results are given in Table 6.5 and displayed in Figure 6.15, and for SAT systems, in Table 6.6 and Figure 6.16. (Note that WER, rather than accuracy, is graphed). In all cases, a single full covariance re-estimation was carried out, and CMLLR transforms from the diagonal models were applied. Estimation of the shrinkage parameter used the globally-pooled version of the shrinkage statistics.

The results demonstrate that off-diagonal smoothing is essential for good performance with full covariance models on the CTS task: the reduction in WER over diagonal models is more than doubled, compared to the naive full covariance systems, when the optimal prior weight is used. The differences, both between the best smoothed systems and the unsmoothed system, and between every full covariance system and the diagonal system, are highly statistically significant. Comparing identically smoothed full covariance systems, the application of SAT generally resulted in weakly significant WER improvement ($p < 5\%$).

With and without SAT, the best results were obtained with an single, appropriately tuned prior weight $\tau$, in this case found to be approximately in the range 80–100. However, the analytic method for obtaining a shrinkage parameter directly from the data was again shown to be effective, achieving close to the best performance obtained by tuning $\tau$ on the test set. Differences in WER between the $\tau = 80$ and $\tau = 100$ smoothed systems and the shrinkage system were not statistically significant.

| Prior $\tau$ | Trigram WER |
|---|---|
| 0 | 32.1 |
| 10 | 31.3 |
| 20 | 31.0 |
| 40 | 30.7 |
| 60 | 30.6 |
| 80 | **30.5** |
| 100 | **30.5** |
| 125 | 30.6 |
| 150 | 30.7 |
| 175 | 30.7 |
| 200 | 30.8 |
| 300 | 31.0 |
| 400 | 31.3 |
| Diagonal | 33.3 |
| Semi tied | 33.1 |
| Shrinkage | 30.6 |

Table 6.5: %WER results for full covariance systems smoothed with a diagonal prior, with varying prior weight, $\tau$.

### 6.2.4 Effects on condition number

In Section 3.1.2 we discussed the importance of having well-conditioned covariance matrices, and in Section 5.1.5, made claims about the benefits of the shrinkage estimator in this regard. We briefly illustrate the effects of off-diagonal smoothing on the condition number of the covariance matrices.

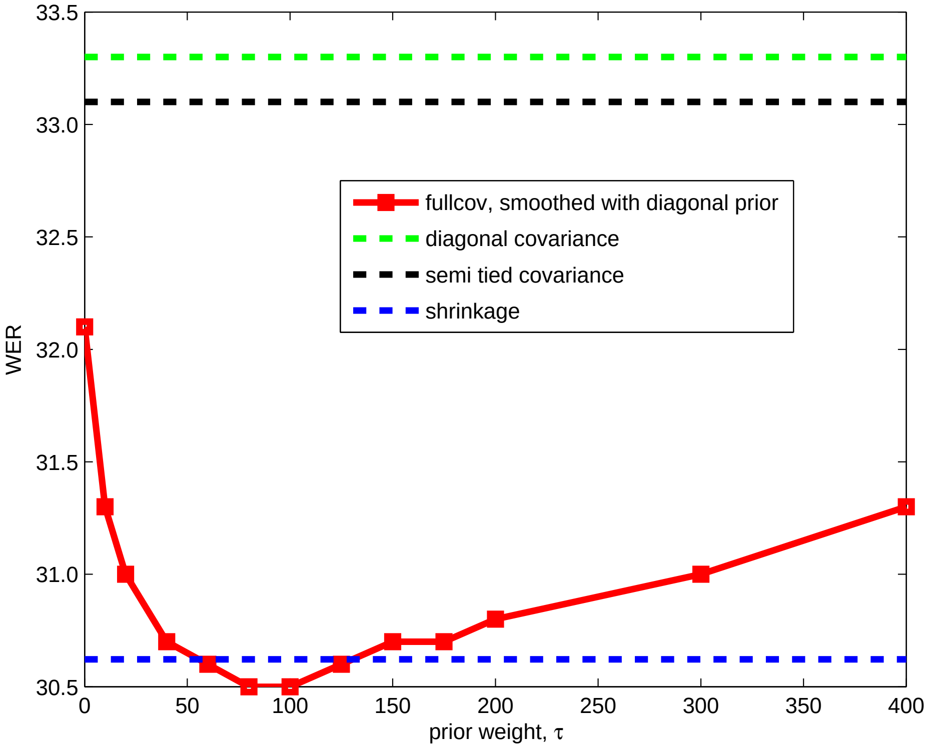

Figure 6.15: %WER results for full covariance systems smoothed with a diagonal prior, with varying prior weight, $\tau$.

| Prior $\tau$ | Trigram WER |
|---|---|
| 0 | 32.1 |
| 10 | 31.1 |
| 20 | 30.9 |
| 40 | 30.5 |
| 60 | 30.4 |
| 80 | **30.3** |
| 100 | 30.4 |
| 125 | 30.4 |
| 150 | 30.4 |
| 175 | 30.5 |
| 200 | 30.6 |
| 300 | 30.9 |
| 400 | 31.1 |
| Diagonal | 32.9 |
| Shrinkage | 30.4 |

Table 6.6: %WER results for full covariance SAT systems, smoothed with a diagonal prior.

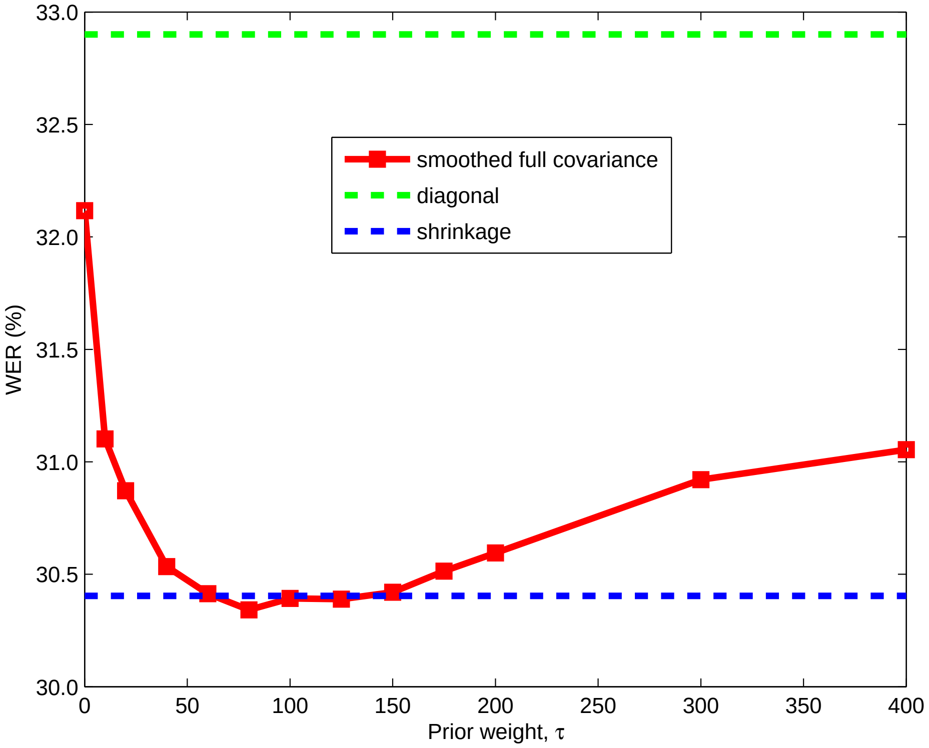

Figure 6.16: %WER results for full covariance SAT systems, with smoothed with a diagonal prior, with varying prior weight, $\tau$.

Figure 6.17 shows the drop in the mean condition number (taken over all Gaussians in the system) as the prior weight is increased. However, the mean condition number is less important to the system performance than a reduction in Gaussians with very high condition number; we therefore also show, in Figure 6.18 the change in the standard deviation of the condition number. There is a dramatic fall in this value when a diagonal prior is used, compared to the use of the naive unsmoothed matrix – this matches the sharp fall in error rate demonstrated in Figures 6.15 and 6.16.

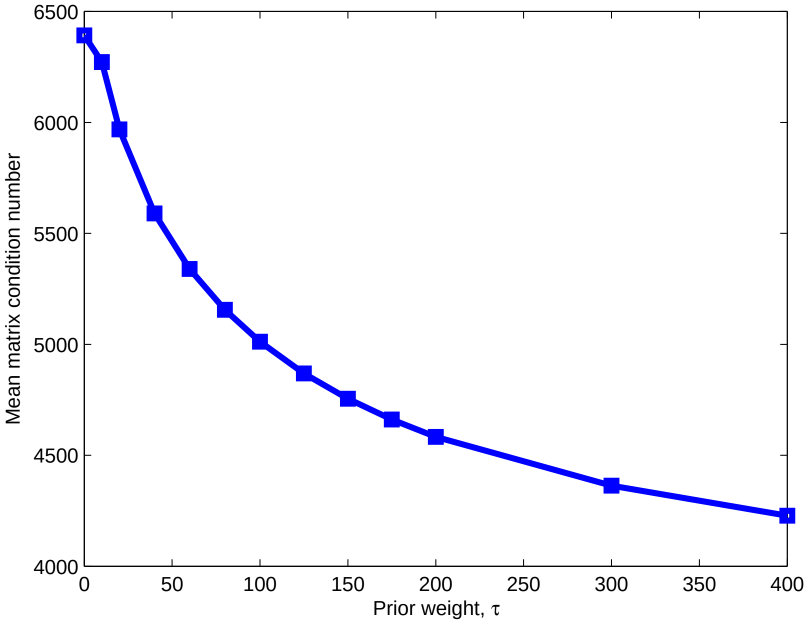

Figure 6.17: Mean covariance condition number across all Gaussians, with varying prior weight, $\tau$.

## 6.3 Summary

In this chapter we have described experimental results with full covariance models on the TIMIT phone recognition task and on a large-vocabulary conversational telephone speech task. On the TIMIT task, we investigated sparse Gaussian graphical models and models smoothed with an off-diagonal prior, varying the

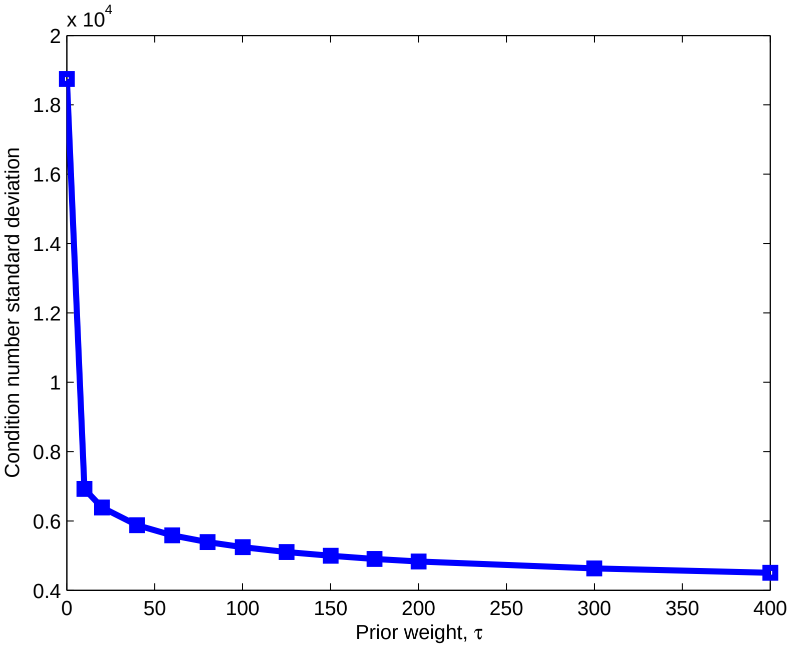

Figure 6.18: Standard deviation of covariance condition number across all Gaussians, with varying prior weight, $\tau$.

quantity of available training data; for the CTS system, we compared smoothed full covariance systems with diagonal models.

Results with the sparse GM systems were not promising. However, we found that off-diagonal smoothing was essential for good performance of the full covariance models, particularly when the quantity of training data was reduced. With appropriate smoothing, the full covariance systems generally outperformed diagonal equivalents by a significant margin. Systems with an analytically obtained shrinkage parameter achieved close to the optimal performance from a range of smoothing weights, but for both tasks the best performance was consistently obtained by an appropriately chosen prior weight, found to be approximately 50 for the TIMIT task and 100 for the CTS task.

# Chapter 7

# Discriminative training methods

## 7.1 Discriminative training criteria

In introducing discriminative training, we return to the formulation of statistical pattern recognition presented in the Introduction. In this simplified account of the speech recognition problem, an input $x$ is classified as class $y$ having the maximum posterior probability:

$$\hat{y} = \arg\max_{y \in \mathcal{C}} p(y|x) \tag{7.1}$$

In this context, $p(y|x)$ is a *discriminant function*. In the traditional approach, we wish to obtain a good approximation to $p(y|x)$, given training data $(x_r, y_r)$. It is possible to approximate $p(y|x)$ directly using, for example, Conditional Random Fields (Lafferty *et al.*, 2001) or Artificial Neural Networks (ANNs) (Bourlard & Morgan, 1994). These approaches are implicitly discriminative. However, for ASR, the high-dimensional nature of both the input space and the output space make a parametric generative modelling approach attractive. The classification becomes

$$\begin{aligned} \hat{y} &= \arg\max_{y \in \mathcal{C}} p(x, y) && (7.2) \\ &\approx \arg\max_{y \in \mathcal{C}} p_\theta(x|y)p(y) && (7.3) \\ &= \arg\max_{y \in \mathcal{C}} D_\theta(x, y) && (7.4) \end{aligned}$$

where $D_\theta(x, y) = \log p_\theta(x|y)P(y)$ is the discriminant function. The subscript $\theta$ indicates the parametrised version of the generative probability, which we take to use the CD-HMM, for its many attractive properties discussed in earlier chapters. The discriminant function is manually specified and the parameters are learned from data; together they form a *learning machine*.

In ML training, we aim to find parameters that "best explain" the observed training data. Given inputs $x_r$ labelled with the correct class $y_r$, We seek $\theta$ giving the highest log-likelihood of the data by maximising the objective function:

$$F_{\mathrm{ML}}(\theta) = \sum_r D_\theta(x_r, y_r) \tag{7.5}$$

We have seen that ML training has deficiencies when training data is limited. Even in the limiting case of infinite training data, the optimality of ML training for classification requires the assumption that the generative model is correct. The simplifying assumptions used in the CD-HMM framework mean, of course, that this is not true.

### 7.1.1 Minimum classification error

The discriminative training approaches we introduce here retain the generative modelling method for computing the discriminant function, but seek to correct for the lack of model-correctness by explicitly considering the classification decisions made by the model. If the correct class for input $x_r$ is $y_r$, it is not the size of $D_\theta(x_r, y_r)$, that is important per se, rather that $D_\theta(x_r, y_r)$ should be larger than $D_\theta(x_r, y)$ for all competing classes, $y \neq y_r$. We define the margin $\mathcal{E}_r$ by

$$\mathcal{E}_r = D_\theta(x_r, y_r) - \max_{y \neq y_r} D_\theta(x_r, y) \tag{7.6}$$

So that the classification decision (7.4) is correct for the $r$th example if $\mathcal{E}_r > 0$. This motivates the Minimum Classification Error (MCE) criterion (Juang & Katagiri, 1992) for training:

$$F_{\mathrm{MCE}}(\theta) = \sum_r H(\mathcal{E}_r) \tag{7.7}$$

where $H(x)$ is the step function

$$H(\mathcal{E}_r) = \begin{cases} 0 & \mathcal{E}_r < 0 \\ 1 & \mathcal{E}_r \geq 0 \end{cases} \tag{7.8}$$

This is non-differentiable, so difficult to optimise, but the step function can be replaced by a sigmoid:

$$H(\mathcal{E}_r) = \frac{1}{1 + e^{-a\mathcal{E}_r}} \tag{7.9}$$

McDermott & Katagiri (2004) show that, subject to the choice of the parameter $a$, the expected classification error using the MCE-trained models converges to the expected error under the true model-free probabilities.

### 7.1.2 Maximum mutual information and conditional maximum likelihood

For ASR, the MCE criterion has the disadvantage that the 0/1 error function for each utterance does not closely match the typical performance metrics for ASR, where we measure the number of words correctly recognised, rather than the number of complete utterances. In particular, the MCE criterion is highly sensitive to how the training data is segmented into utterances. Consequently, MCE training has most commonly been applied to isolated word recognition. An alternative criterion used more widely for large vocabulary ASR is the Maximum Mutual Information (MMI) criterion (Bahl *et al.*, 1986). The objective is to maximise the estimated mutual information between the acoustic features and the transcriptions:

$$\begin{aligned} F_{\text{MMI}}(\theta) &= \sum_r \log \frac{p(x_r, y_r)}{p(x_r)P(y_r)} && (7.10) \\ &= \sum_r \left[\log \frac{p(x_r, y_r)}{p(x_r)} - \log P(y_r)\right] && (7.11) \\ &= \sum_r \left[\log \frac{p_\theta(x_r|y_r)P(y_r)}{\sum_y p_\theta(x_r|y)P(y)} - \log P(y_r)\right] && (7.12) \end{aligned}$$

If the prior probabilities $P(y)$ are held constant (in other words, if the language model is fixed), maximising $F_{\text{MMI}}(\theta)$ is equivalent to maximising the sum of the first terms above. This is the conditional maximum likelihood criterion (Nádas, 1983)

$$F_{\text{CML}}(\theta) = \sum_r \log P(y_r|x_r) = \sum_r \log \frac{p_\theta(x_r|y_r)P(y_r)}{\sum_y p_\theta(x_r|y)P(y)} \tag{7.13}$$

Henceforth we assume the priors are fixed and treat the two criteria as equivalent (we usually denote them by MMI). Again, it can be shown (Bouchard & Triggs, 2004; Schlüter & Ney, 2001) that that expected error rate using MMI-trained models converges to the model free expected error rate in the limit of infinite training data.

The use of the MMI/CML criteria may also be motivated from a margin perspective. Suppose we have the goal of minimising the margin by which utterances are incorrectly classified. We redefine the margin $\mathcal{E}_r$ to include the correct hypothesis in the second term,

$$\mathcal{E}_r = D_\theta(x_r, y_r) - \max_y D_\theta(x_r, y) \tag{7.14}$$

so that $\mathcal{E}_r = 0$ for a correctly classified utterance, $\mathcal{E}_r < 0$ otherwise. Then the objective is to maximise

$$F(\theta) = \sum_r \mathcal{E}_r \tag{7.15}$$

The maximisation in the second term can be approximated by a soft upper bound:

$$\max_y D_\theta(x_r, y) \leq \log \sum_r e^{D_\theta(x_r,y)} \tag{7.16}$$

and therefore a lower bound on the objective function (7.15) is given by:

$$\begin{aligned} F(\theta) &= \sum_r \Big[ D_\theta(x_r, y_r) - \log \sum_y e^{D_\theta(x_r,y)} \Big] && (7.17) \\ &= \sum_r \Big[ \log p_\theta(x_r|y_r)P(y_r) - \log \sum_y p_\theta(x_r|y)P(y) \Big] && (7.18) \\ &= \sum_r \log \frac{p_\theta(x_r|y_r)P(y_r)}{\sum_y p_\theta(x_r|y)P(y)} = F_{\mathrm{CML}}(\theta) && (7.19) \end{aligned}$$

In this thesis we use the MMI criterion for discriminative training. Returning to the notation of Chapter 2, where a training utterance $r$ has word transcription $W_r$ and acoustic observations $O_r$, the MMI objective function is given by

$$\begin{aligned} F_{\mathrm{MMI}}(\theta) &= \sum_r \log p(W_r|O_r, \theta) && (7.20) \\ &= \sum_r \log \frac{p(O_r|W_r, \theta)^\kappa P(W_r)}{\sum_W p(O_r|W, \theta)^\kappa P(W)} && (7.21) \end{aligned}$$

The denominator is a sum over all possible word sequences. $\kappa$ is a scaling factor set to $\nu^{-1}$, the inverse of the language model scaling factor used during decoding, to compensate for the fact that the acoustic model simplifications lead to them overestimating the probability.

### 7.1.3 Minimum Bayes risk

Minimum Bayes risk training (Doumpiotis & Byrne, 2004; Kaiser *et al.*, 2000) is a general discriminative training framework whereby we seek parameters to minimise the expected posterior loss on the training set. The general form of the objective function is

$$F_{\mathrm{MBR}}(\theta) = -\sum_r \sum_w P(W|O_r, \theta) L(W_r, W) \tag{7.22}$$

where $L(W_r, W)$ is a loss function measuring the error between the hypothesised sentence $W$ and the correct sentence $W_r$ (typically it is computed using the Levenstein distance). This aims to more closely optimise the WER by which ASR performance is measured. In this category, the Minimum Word Error (MWE) criterion attempts to minimise the expected word error directly; a more commonly-used alternative is the Minimum Phone Error (MPE) criterion (Povey & Woodland, 2002). where the loss function is computed a the phone level. This gives a greater ability to generalise to test data.

To obtain good test set performance[1] with MPE-trained models Povey & Woodland (2002) found it necessary to smooth the parameter updates with values of the parameters obtained using both ML and MMI criteria. Although MPE is used in many state-of-the-art large-vocabulary ASR systems, we elected not to use MPE training for the discriminative experiments reported below, because the interaction between the various smoothing constants tends to obscure the experimental analysis of full covariance smoothing.

## 7.2 Objective function maximisation

### 7.2.1 MMI auxiliary functions

A method for training CD-HMM parameters using the MMI objective function was developed by Normandin & Morgera (1991). From

$$F_{\mathrm{MMI}}(\theta) = \log p(O_r|W_r, \theta)^{\kappa} P(W_r) - \log \sum_W p(O_r|W, \theta)^{\kappa} P(W) \tag{7.23}$$

[1] We discuss the generalisation abilities of discriminatively-trained models in a later section

(the sum over utterances $r$ is dropped for clarity). Given some initial parameter set $\theta^0$, it can be shown, using a similar procedure to that in Section 2.3.2, that the function

$$G^{\text{num}}(\theta, \theta^0) = \sum_Q \sum_M P(Q, M|O, \theta^0, W_r)^\kappa P(W_r) \log p(O, Q, M|\theta, W) \tag{7.24}$$

is a lower bound for the first term in (7.23), with equality at $\theta = \theta^0$. Similarly,

$$G^{\text{den}}(\theta, \theta^0) = \sum_Q \sum_M \sum_W P(Q, M|O, \theta^0, W_r)^\kappa P(W_r) \log p(O, Q, M|\theta, W) \tag{7.25}$$

is a lower bound for the second term, again with equality at $\theta = \theta^0$. The two functions are respectively known as the numerator and denominator auxiliary functions. Re-expressing the sums as over the frames $t$, and ignoring terms that do not vary with the Gaussian parameters, we obtain:

$$G^{\text{num}}(\theta, \theta^0) = \sum_t \sum_j \sum_m \gamma_{jm}^{\text{num}}(t) \log f_{jm}(o_t) \tag{7.26}$$

$$G^{\text{den}}(\theta, \theta^0) = \sum_t \sum_j \sum_m \gamma_{jm}^{\text{den}}(t) \log f_{jm}(o_t) \tag{7.27}$$

where $\gamma_{jm}^{\text{num}}(t)$ and $\gamma_{jm}^{\text{den}}(t)$ are the occupancy probabilities given the correct transcription, and over all possible transcriptions, respectively. The probabilities can be found using the forward-backward algorithm. In the case of large vocabulary ASR, clearly summing over all possible word transcriptions is intractable. A solution was proposed by Povey & Woodland (2000): for the denominator probabilities, an existing set of models is used to perform recognition on the training data, generating a lattice for each utterance containing the most closely competing potential transcriptions. The forward-backward probabilities are computed over the arcs of these lattices. The numerator probabilities can also be computed using lattices that match the known transcription but contain multiple alignment hypotheses.

### 7.2.2 Parameter updates

Using the two auxiliary functions, we define

$$G(\theta, \theta^0) = G^{\text{num}}(\theta, \theta^0) - G^{\text{den}}(\theta, \theta^0) \tag{7.28}$$

Whilst the functions $G^{\text{num}}(\theta, \theta^0)$ and $G^{\text{den}}(\theta, \theta^0)$ are lower bounds for their respective terms in the objective $F_{\text{MMI}}(\theta)$, their difference is not a lower bound for the objective, due to the subtraction. However, at $\theta^0$ the gradients are equal:

$$\left.\frac{\partial F_{\text{MMI}}(\theta)}{\partial \theta}\right|_{\theta^0} = \left.\frac{\partial G(\theta, \theta^0)}{\partial \theta}\right|_{\theta^0} \tag{7.29}$$

$G(\theta, \theta^0)$ is known as a *weak sense* auxiliary function for $F_{\text{MMI}}(\theta)$: an increase in the value of the auxiliary will increase the objective, in a region close to $\theta^0$.

A variety of schemes have been proposed for using the weak sense auxiliary for maximising the MMI objective. The principal concerns are that the updated parameters are valid (in particular, covariances must be positive definite) and that the update is sufficiently small that the auxiliary remains a good approximation to the objective. Schemes include making the denominator term in the auxiliary function a linear function of the covariance (Povey, 2003) and using a line search with convex constraints (Liu & Soong, 2008; Liu *et al.*, 2007). We investigated full covariance updates using Newton's method with convex constraints on the TIMIT task (Bell & King, 2008). However, for the large vocabulary experiments presented below, we used the smoothing technique proposed by Normandin & Morgera (1991), with refinements by Povey & Woodland (2000). This is known as the extended Baum-Welch (EBW) algorithm.

We maximise a smoothed version of the auxiliary given by

$$G(\theta, \theta^0) = G^{\text{num}}(\theta, \theta^0) - G^{\text{den}}(\theta, \theta^0) + G^{\text{sm}}(\theta, \theta^0) \tag{7.30}$$

where $G^{\text{sm}}(\theta, \theta^0)$ is a smoothing function with the same functional form as the other two functions, with a a maximum at $\theta = \theta^0$, so that the gradient of the auxiliary at $\theta^0$ is unaffected by its addition.

We adopt the notation of Sim & Gales (2006) for discriminative training with full covariance. Equation 7.30 can be expressed as a sum over all Gaussians. We describe the optimisation of the parameters of Gaussian $m$ (dropping the dependence on $j$), so consider only those terms. In Section 3.1.1 we wrote the log-density of a Gaussian $m$ as

$$\sum_t \gamma_m(t) \log f_m(o_t) = \sum_t \gamma_m(t) \Big[ -\frac{1}{2} \log |\Sigma_m| - \frac{1}{2}(o_t - \mu_m)^T \Sigma_m^{-1} (o_t - \mu_m) \Big] \tag{7.31}$$

In discriminative training, $\mu_m$ is no longer simply set to the sample mean, so it is more helpful write the density (7.31) in terms of non-centralised statistics:

$$\sum_t \gamma_m(t)\Big[-\frac{1}{2}\log|\Sigma_m| - \frac{1}{2}\operatorname{tr}\Sigma_m^{-1}\sum_t \gamma_m(t)(o_t o_t^T - o_t\mu_m^T - \mu_m o_t^T + \mu_m\mu_m^T)\Big] \tag{7.32}$$

$$= -\frac{1}{2}\Big[\beta_m \log|\Sigma_m| + \operatorname{tr}\Sigma_m^{-1}(Y_m - x_m\mu_m^T - \mu_m x_m^T + \mu_m\mu_m)\Big] \tag{7.33}$$

with statistics $\beta_m$, $x_m$ and $Y_m$. In the MMI case, the statistics for $G^{\text{num}}(\theta,\theta^0)$ are given by

$$\beta^{\text{num}} = \sum_t \gamma_m^{\text{num}}(t), \quad x_m^{\text{num}} = \sum_t \gamma_m^{\text{num}}(t)o_t, \quad Y_m^{\text{num}} = \sum_t \gamma_m^{\text{num}}(t)o_t o_t^T \tag{7.34}$$

and the statistics for $G^{\text{den}}(\theta,\theta^0)$ by

$$\beta^{\text{den}} = \sum_t \gamma_m^{\text{den}}(t), \quad x_m^{\text{den}} = \sum_t \gamma_m^{\text{den}}(t)o_t, \quad Y_m^{\text{den}} = \sum_t \gamma_m^{\text{den}}(t)o_t o_t^T \tag{7.35}$$

The combined statistics, for $G^{\text{num}}(\theta,\theta^0) - G^{\text{den}}(\theta,\theta^0)$ are then

$$\beta_m^c = \beta_m^{\text{num}} - \beta_m^{\text{den}}, \quad x_m^c = x_m^{\text{num}} - x_m^{\text{den}}, \quad Y_m^c = Y_m^{\text{num}} - Y_m^{\text{den}} \tag{7.36}$$

The statistics for the smoothing term are derived from the original parameters $(\mu_m^0, \Sigma_m^0)$, and are given by

$$\beta^{sm} = D_m \tag{7.37}$$

$$x_m^{sm} = D_m\mu_m^0 \tag{7.38}$$

$$Y_m^{sm} = D_m(\Sigma_m^0 + \mu_m^0\mu_m^{0\,T}) \tag{7.39}$$

$D_m$ is known as smoothing constant. The terms of 7.30 including Gaussian $m$ can be expressed as

$$-\frac{1}{2}\Big[(\beta_m^c{+}\beta_m^{sm})\log|\Sigma_m|{+}\operatorname{tr}\Sigma_m^{-1}(Y_m^c{+}Y_m^{sm}{-}(x_m^c{+}x_m^{sm})\mu_m^T{-}\mu_m(x_m^c{+}x_m^{sm})^T{+}\mu_m\mu_m^T)\Big] \tag{7.40}$$

Then $G(\theta,\theta^0)$ is maximised by setting

$$\hat{\mu}_m = \frac{x_m^c + x_m^{sm}}{\beta_m^c + \beta_m^{sm}} \tag{7.41}$$

$$\hat{\Sigma}_m = \frac{Y_m^c + Y_m^{sm}}{\beta_m^c + \beta_m^{sm}} - \hat{\mu}_m\hat{\mu}_m^T \tag{7.42}$$

It can be seen that the larger the magnitude of the smoothing constant, the smaller the size of the parameter updates.

### 7.2.3 Choosing the smoothing constant

Povey & Woodland (2000) proposed setting the smoothing constants on a per-Gaussian level, and suggested setting them at twice the level necessary to ensure positive variances, floored at twice the denominator occupancy, $\sum_t \gamma_{jm}^{\text{den}}(t)$. In the full covariance case the equivalent is to ensure that the covariance matrix is positive definite. Equation 7.42 can be expressed as:

$$\hat{\Sigma}_m = \frac{Y_m^c + Y_m^{sm}}{\beta_m^c + \beta_m^{sm}} - \frac{(x_m^c + x_m^{sm})(x_m^c + x_m^{sm})^T}{(\beta_m^c + \beta_m^{sm})^2} \tag{7.43}$$

$$= \frac{(Y_m^c + Y_m^{sm})(\beta_m^c + \beta_m^{sm}) - (x_m^c + x_m^{sm})(x_m^c + x_m^{sm})^T}{(\beta_m^c + \beta_m^{sm})^2} \tag{7.44}$$

Separating this into terms that are quadratic, linear and constant in $D_m$, we have

$$\hat{\Sigma}_m = B_0 + D_m B_1 + D_m^2 B_2 \tag{7.45}$$

where

$$B_0 = \beta_m^c Y_m^c - x_m^c {x_m^c}^T \tag{7.46}$$

$$B_1 = Y_m^c + \beta_m^c Y_m^{sm} - x_m^c {\mu_m^0}^T - \mu_m^0 {x_m^c}^T \tag{7.47}$$

$$B_2 = \Sigma_m^{sm} \tag{7.48}$$

We require

$$B_0 + D_m B_1 + D_m^2 B_2 \succ 0 \tag{7.49}$$

Which can be ensured by setting $D_m$ to the largest solution of the quadratic eigenvalue problem

$$|B_0 + \lambda B_1 + \lambda^2 B_2| = 0 \tag{7.50}$$

We discuss the solution of this in Appendix A.1. In their full covariance system, Chen *et al.* (2006) avoid the need for this by using an iterative procedure: choosing a value for $D_m$, checking whether the resulting update yields a positive definite matrix, and doubling $D_m$ if not. For precision matrix subspace methods, if only basis coefficients are updated then $D_m$ can be found in a memory-efficient manner by solving a quadratic equation for each dimension (see Sim & Gales, 2006).

The situation is even simpler if we update the variances without updating the means. Then

$$\hat{\Sigma}_m = \frac{Y_m^c + Y_m^{sm}}{\beta_m^c + \beta_m^{sm}} - \mu_m^0 \mu_m^{0\ T} \tag{7.51}$$

$$= \frac{Y_m^c - \beta_m^c \mu_m^0 \mu_m^{0\ T}}{\beta_m^c + \beta_m^{sm}} \tag{7.52}$$

and we require

$$B_0 + D_m B_1 \succ 0 \tag{7.53}$$

where $B_0 = Y_m^c - \beta_m^c \mu_m^0 \mu_m^{0\ T}$, $B_1 = \Sigma^0$. This can be guaranteed by setting $D_m$ to the largest eigenvalue of $B_0 B_1^{-1}$.

### 7.2.4 Parameter initialisation

The standard EM algorithm guarantees convergence to at least a local optimum of the ML objective function; it is not possible to obtain such convergence properties for the MMI auxiliary function. As we have seen, it is necessary to restrict the size of parameter updates to ensure an increase in the objective. For this reason a good choice of initial parameters is important. The standard practice is to initialise the EBW algorithm with ML-trained parameters, obtained using the EM algorithm.

## 7.3 Generalisation of discriminative estimators

As discussed earlier, when using full covariance models with many parameters, it is particularly important to consider the generalisation performance. The model-free optimality properties of the discriminative estimators discussed above do not guarantee good generalisation when the amount of training data is limited. In this section we discuss standard methods used for improving the generalisation of MMI models, before describing large-margin techniques that have been more recently investigated. We go on to consider the generalisation of full covariance models within a discriminative framework, when lower dimensional smoothing priors are used.

Seemingly different methods for improving generalisation in a discriminative setting can sometimes be shown to be broadly equivalent, and, since the shrinkage-based covariance refinements we introduce are largely motivated by this goal, we feel that this merits the detailed analysis presented here.

### 7.3.1 Increasing confusable data

Generalisation ability is of course increased if the amount of training data is increased. For a fixed amount of training data, the generalisation of discriminative estimators can be increased by increasing the number of acoustically confusable examples (those contributing to the posterior $\gamma^{\text{den}}$). As explained by Povey & Woodland (2001), this can be achieved by weakening the language model used for the forward-backward computations over the lattices. Typically, a unigram LM is used. Similarly, it is the desire to increase the confusable examples which motivates the use of $\kappa$, the inverse LM scaling factor, when computing $\gamma^{\text{den}}$ using the forward-backward algorithm, rather than scaling-up the language model probabilities.

### 7.3.2 I-smoothing

ML trained parameters are known to have lower variance than MMI-trained parameters (Bouchard & Triggs, 2004; Nádas, 1983), and several authors have proposed using a linear combination of the two to improve generalisation performance. An example used for ASR is the *H-criterion* of Gopalakrishnan *et al.* (1988). Povey & Woodland (2002) proposed weighting the linear combination according to the quantity of training data available, so that in the limiting case of infinite training data, the MMI criterion is used. This is known as *I-smoothing*. To $G(\theta, \theta^0)$ is added additional prior term $G^I(\theta, \theta^0)$ with weight $\tau^I$ and statistics proportional to the ML (numerator) statistics:

$$\beta_m^I = \tau^I \tag{7.54}$$

$$x_m^I = \tau^I \frac{x_m^{\text{num}}}{\beta_m^{\text{num}}} \tag{7.55}$$

$$Y_m^I = \tau^I \frac{Y_m^{\text{num}}}{\beta_m^{\text{num}}} \tag{7.56}$$

More recently a similar smoothing effect has been achieved by instead smoothing towards the previous parameter values by incrementing the value of $D_m$; this has been shown to yield slightly improved performance over the earlier technique (Povey *et al.*, 2008).

### 7.3.3 Large-margin estimation

Vapnik (1995) introduced Support Vector Machines (SVMs). These are binary linear classifiers shown to possess optimality properties with regards to generalisation ability. The SVMs are trained to maximise the margin of correct classification of the training data, and it can be shown (Burges, 1998) that under certain conditions, the size of the margin controls a bound on the generalisation ability. Whilst binary classification techniques are not suitable for large vocabulary ASR, these results have motivated the use of large margin estimation for discriminative training for ASR, as a means of controlling the generalisation ability of the models.

For an ASR system, the margin may be defined with reference to the discriminant functions. Returning to the notation of earlier in this Chapter, defining the margin for an utterance $r$ by

$$\mathcal{E}_r = D_\theta(x_r, y_r) - \max_{y \neq y_r} D_\theta(x_r, y) \tag{7.57}$$

a canonical large-margin objective function to maximise would be

$$F_{\text{LM}}(\theta) = \min_r \mathcal{E}_r \tag{7.58}$$

or equivalently

$$F_{\text{LM}}(\theta) = \max \rho, \quad \mathcal{E}_r > \rho \tag{7.59}$$

However, this is not a bounded problem; nor is it tractable. Efforts to incorporate large-margin techniques into CD-HMM parameter estimation have focused on approximations to the objective function for which a solution is feasible. Jiang *et al.* (2006); Li & Jiang (2006) restrict, at each iteration, the set of utterances used in (7.58) to those for which $0 \leq \mathcal{E}_r \leq \epsilon$ with some preset $\epsilon$, and recast the parameter updates as a constrained convex optimisation problem.

However, this excludes misclassified utterances from contributing to the objective function. Li *et al.* (2006) proposed soft-margin estimation (SME). Here, the separation $\rho$ is chosen heuristically, and all utterances for which the margin, $\mathcal{E}_r$, of correct classification is less than $\rho$ are included:

$$F_{\text{SME}}(\theta) = \sum_r \mathcal{E}_r, \quad \mathcal{E}_r < \rho \tag{7.60}$$

In this work, the margin was normalised by by the number of frames in the utterance to avoid longer utterances naturally having higher separation. SME does, however, suffer from sensitivity to outliers in the training data that are misclassified by a large margin, since these are able to dominate (7.60). This can be remedied (Yu *et al.*, 2008) by replacing the linear function in (7.60) by a step function (or a sigmoid, its soft equivalent) as in MCE – this is known as large-margin MCE (LM-MCE):

$$F_{\text{LM-MCE}}(\theta) = \sum_r H(\mathcal{E}_r - \rho) \tag{7.61}$$

with $H(x)$ as in (7.8) or (7.9).

As we discussed with reference to MCE earlier, the methods described above do not readily extend to sequence classification as required for ASR – indeed, they have largely been employed for isolated digit recognition. This disadvantage motivated Sha & Saul (2007) to propose a margin scaled by the (frame-wise) Hamming distance $\mathcal{H}$ between the correct sequence and the hypothesis. Rather than penalising $\mathcal{E}_r - \rho < 0$ in the objective function, we apply a penalty when

$$\min_{y \neq y_r} \left[ D_\theta(x_r, y_r) - D_\theta(x_r, y) - \rho\mathcal{H}(y, y_r) \right] < 0 \tag{7.62}$$

We use the left-hand side to define a new margin function

$$\begin{aligned} \mathcal{E}_r^{\text{H}}(\rho) &= \min_{y \neq y_r} \left( D_\theta(x_r, y_r) - D_\theta(x_r, y) - \rho\mathcal{H}(y, y_r) \right) && (7.63) \\ &= D_\theta(x_r, y_r) - \max_{y \neq y_r} \left[ D_\theta(x_r, y) - \rho\mathcal{H}(y, y_r) \right] && (7.64) \end{aligned}$$

giving a large-margin objective

$$\sum_r \mathcal{E}_r^{\text{H}}(\rho), \quad \mathcal{E}_r^{\text{H}}(\rho) < 0 \tag{7.65}$$

Saon & Povey (2008) showed that using a smoothed version of this margin leads to the large-margin objective function being expressible as a simple modification of the standard MMI objective, making it suitable for large-vocabulary systems. We can incorporate the correct transcription into the maximisation in (7.64) without affecting the objective. Replacing the maximisation with a soft upper bound as in Equation 7.16 and noting that the constraint $\mathcal{E}_r^{\mathrm{H}}(\rho) < 0$ is then automatically satisfied, yields the objective function

$$F_{\text{LM-MMI}}(\theta) = \sum_r \Big[ D_\theta(x_r, y_r) - \log \sum_y e^{D_\theta(x_r,y)+\rho\mathcal{H}(y,y_r)} \Big] \tag{7.66}$$

This function can be maximised by a relatively straightforward modification to the procedure described in Section 7.2. When computing the forward-backward probabilities over the denominator lattices, used to compute the $\gamma^{\text{den}}$ posteriors, the acoustic log-likelihood of each are increased by the contribution of that arc to the total Hamming distance for the transcription.

Throughout the analysis in this Section we have presented $\rho$ as some preset constant. A larger margin is more desirable (although not in the limit of infinite training data), and so we might more properly treat $\rho$ as a variable, present a large-margin objective as a function of $\rho$. We would then attempt to find

$$\arg\max_{\theta,\rho} \big[\rho + \lambda F_{\text{LM}}(\theta, \rho)\big] \tag{7.67}$$

for some scale factor $\lambda$. In fact this is not particularly important, since the optimal $\rho$ is dependent on the choice of $\lambda$, and the joint optimisation is typically achieved by selecting $\rho$ from a limited set of possible values. In practice, we can view it as hyperparameter that must be tuned using test data.

## 7.4 Full covariance modelling with MMI

We now focus on the use of discriminative techniques for estimating full covariance parameters. The experimental results presented in Chapter 6 showed that when using full covariance models, a shrinkage estimator outperforms the standard sample covariance matrix, with the effects more pronounced when training data is limited. Put another way, off-diagonal smoothing – or use of a diagonal covariance prior – is necessary for good performance on test data. We explained how the shrinkage estimator optimises a bias/variance trade-off to obtain good generalisation performance.

However, the approach adopted was explicitly generative: the estimators were essentially trained using the ML criterion, and the the analytically-obtained shrinkage parameter was chosen to minimise the expected deviation from some true unobserved covariance matrix. In reality, any models used are far from being a correct model for speech, and there are no "true" matrices; in this section we ask whether similar techniques can be used to improve the performance of full covariance models when the aim is to maximise a discriminative criterion. The analysis presented here is somewhat empirically-based; however, we use it as a motivation for the "recipes" for full covariance discriminative training that we investigate on the CTS recognition task in the following chapter.

### 7.4.1 Model correctness

As an illustration, we return to the artificial three-way classification problem presented as a motivating example in the introduction to Chapter 3. Data for each class was generated by a full covariance Gaussian. Recall that when a large number of samples was used for training with ML, diagonal covariance models achieved an error rate of 5.3%, whilst full covariance models achieved an error rate of 1.2%. Our first observation is that applying MMI training improves the performance of the weaker models: after 12 iterations of the EBW algorithm, the error rate is reduced to 1.6%. The decision boundaries from the ML-trained and MMI-trained diagonal models are compared in Figure 7.1. Applying MMI training to the full covariance models does not result in a performance improvement. This illustrates the principle that MMI training corrects for the invalidity of model correctness assumptions; when the model *is* correct, as in the case of the full covariance models in this setting, there is no advantage to be gained.

### 7.4.2 Discriminative bias and variance

For our purposes, the simulations in the previous section are unrealistic due to the use of a very large number of training samples. As in the general asymptotic analysis in Chapter 5, we instead consider the case where, closer to the reality in ASR, the number of training samples for each Gaussian is of the same order as the dimensionality. A justification for this approach may be found in our CTS system: approximately 10,000,000 frames of training data are shared over 120,000

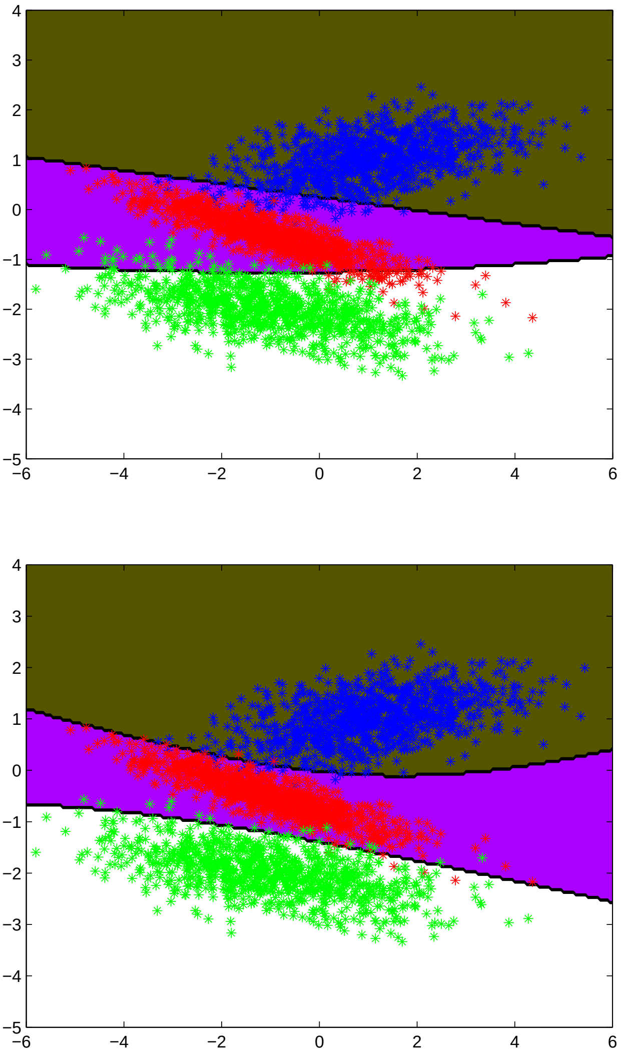

Figure 7.1: Decision boundaries obtained using diagonal covariance models, trained using ML (top) and MMI (bottom)

39-dimensional Gaussians, giving an average of approximately 2.1 samples per Gaussian per dimension. In this situation, if we consider the training data to be randomly sampled from the true distribution, then the estimator will have high variance.

When comparing the performance of estimators of differing dimensionality in Section 5.1.3, we used the notion of a trade-off between bias (in the lower dimensional case) and variance (in the high-dimensional case). We seek to extend the concepts of bias and variance to the discriminative setting. We adopt the bias/variance decomposition of Domingos (2000) (see also Valentini & Dietterich, 2004).

The situation is more complicated than the generative case because we need to define expectations over both the training and the test data. Define $\mathcal{T}_R$ to be a random training set containing $R$ labelled training examples $(x_r, y_r)$ drawn from $p(x, y)$. The model parameters learned from this training set are denoted by $\theta(\mathcal{T}_R)$. Given new unseen data $(x_t, y_t)$, also drawn from $p(x, y)$, the classification decision made by this learning machine is given by

$$\hat{y}(x_t) = \arg\max_y D_{\theta(\mathcal{T}_R)}(x_t, y) \tag{7.68}$$

It is a function of the random $\mathcal{T}_R$ and the random test data $x_t$. The expected classification loss is given by

$$\mathbb{E}_{p(x_t,y_t)}[\mathbb{E}_{\mathcal{T}_R} L(y_t, \hat{y})] \tag{7.69}$$

where $L(y_A, y_B)$ is the loss incurred by classifying $y_A$ as $y_B$. The inner expectation is over the training data; the outer expectation is over the test data.

An optimal learning machine, minimising the expected loss under the true distribution, would classify test data $x_t$ as

$$y^*(x_t) = \arg\min_y \mathbb{E}_{p(y|x_t)} L(y, y_t) \tag{7.70}$$

which corresponds to the optimal Bayes classifier. The *main prediction*, $y_m$, associated with input $x_t$ is the class that would lead to the best performance from the learning machine, averaged over all training data:

$$y_m(x_t) = \arg\min_y \mathbb{E}_{\mathcal{T}_R} L(y, \hat{y}) \tag{7.71}$$

Domingos (2000) defines the "bias" of the classifier to be the loss between this average prediction, and the optimal prediction:

$$\text{bias}(x_t) = L(y_m, y^*) \tag{7.72}$$

whilst the "variance" of the classifier can be defined as average loss relative to the main prediction:

$$\text{var}(x_t) = \mathbb{E}_{\mathcal{T}_R} L(y_m, \hat{y}) \tag{7.73}$$

All these functions are random quantities of the test data. Domingos (2000) showed that for a general loss function, the error of the classifier (7.69) can be decomposed as

$$\mathbb{E}_{p(x_t,y_t)}[\mathbb{E}_{\mathcal{T}_R} L(y_t, \hat{y})] = \mathbb{E}_{p(x_t,y_t)}[c_1 \mathbb{E}_{\mathcal{T}_R}[L(y_m, \hat{y})] + L(y_m, y^*) + c_2 L(y^*, y_t)] \tag{7.74}$$

$$= \mathbb{E}_{p(x_t,y_t)}[c_1 \,\text{var}(x_t) + \text{bias}(x_t) + c_2 N(x_t, y_t)] \tag{7.75}$$

with the values of the constants $c_1$ and $c_2$ depending on the loss function used. The final term is a noise term, representing the loss that is unavoidable even if the optimal classifier is used, due to classes overlapping in feature space.

In practice the true distribution is unknown, so it is not possible to analytically obtain expectations over all possible training sets as required for the decomposition (7.75). Given a limited labelled training set, an approximate procedure suggested by Valentini & Dietterich (2004) is to construct multiple training sets of the same size by sampling with replacement from the full set. For each set, a learning machine is constructed; the test set performance is approximated by evaluating the learning machine on the remaining training examples (approximately $e^{-1}$ of the full training set, on average).

The procedure is not feasible for large-scale ASR systems. We limit our analysis here to the artificial classification task described earlier, and do not carry out cross-validation. This has the advantage that multiple training sets can be easily sampled from the true distribution. We estimated the expectation over training sets using 500 sample sets, and estimated the error of the resulting models using a test set of 3000 samples. Figure 7.2 shows the bias-variance decomposition using the 0/1 loss function on this task, with a varying quantity of training data used to train the Gaussian models (the noise term is constant and included as part

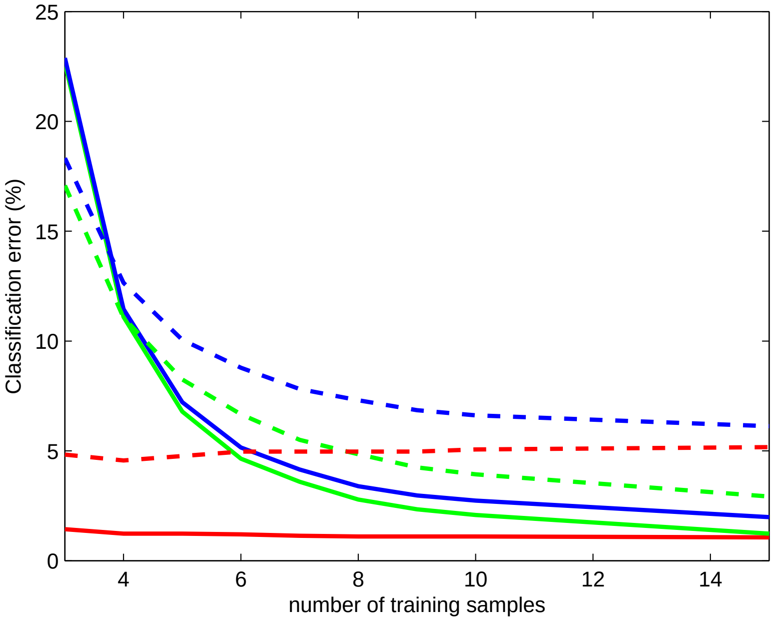

Figure 7.2: Classification error (blue) decomposed into variance (green) and bias (red), for three-class simulated data, with varying number of training samples per class. Solid lines show full covariance models; dashed lines show diagonal covariance models.

of the bias). The performance of full covariance and diagonal covariance models (both ML-trained) is compared.

The graph shows that this measure of variance conforms to the expected properties of a variance measure, in that it reduces with the number of samples, whilst the bias remains approximately constant. It can be seen that the diagonal models have a higher bias than the full covariance models. With just three samples, the high variance of the full covariance models leads to a higher classification error, despite the lower bias. For higher number of samples the variance of the full covariance models falls to slightly lower than that of the diagonal models, contrary to what we would expect from the traditional measure. This can be explained by the fact that the variance about a bias $y_m$ tends to be higher than the variance about an unbiased $y_m$.

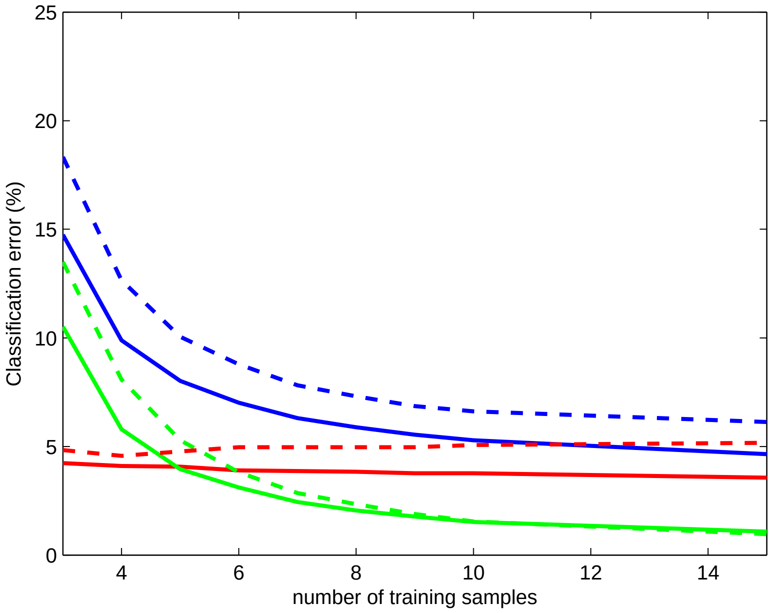

Figure 7.3: Error, variance and bias (coloured as in Figure 7.2) comparing ML-trained diagonal models (dashed) with MMI-trained diagonal models (solid)

In Figure 7.3, we show a similar graph comparing ML-trained and MMI-trained diagonal models. The MMI-trained models yield consistent improvement – this is primarily due to a reduction in bias, rather than a change in variance.

MMI training reduces the error due to the over-simplicity of the model, but does not improve generalisation.

### 7.4.3 Estimation with a diagonal prior

We now analyse the performance of the shrinkage estimator on the artificial data, using the new bias-variance decomposition. We present results here for 3 and 5 samples per class, noting from Figure 7.2 the fact that the diagonal and full covariance models are closest in classification error when the number of samples is in this range. Figure 7.4 (solid lines) shows the effect of varying the shrinkage parameter $\alpha$, interpolating between the full sample covariance matrix ($\alpha = 0$) and a diagonal version ($\alpha = 1$).

It can be seen that there is a steady increase in the bias as $\alpha$ is increased towards the diagonal model. However, initially this is more than offset by the sharp reduction in variance as $\alpha$ is increased from zero. As we found in the experiments on speech data in Chapter 6, the lowest mean error is considerably smaller than the error of both the diagonal and standard full covariance models, this being due to the reduction in variance. The optimal $\alpha$ is considerably lower than value minimising the parameter MSE, which occurs at $\alpha = 0.5$ for $n = 3$ and $\alpha = 0.3$ for $n = 5$.

We propose to improve the performance of the shrinkage estimator by replacing the ML-trained diagonal prior with an MMI-trained diagonal prior. As we have showed in Figure 7.3, the MMI-trained diagonal models have a lower bias than the ML-trained models, and no higher variance. By using this prior, we would hope to maintain the improved error reduction caused by the lower variance, whilst gaining a further reduction by backing off to models with lower bias.

We denote the mean and variance of the MMI-trained diagonal models by $\mu_m^D$, $\Sigma_m^D$. Since the means of the two model sets may now be different, the estimation formulae are slightly changed. Recall from Section 5.1.4 the a smoothed covariance matrix may be expressed equivalently as

$$U_m = (1 - \alpha)S_m + \alpha D_m \equiv \frac{S_m}{\beta_m + \tau} + \frac{\tau D_m}{\beta_m + \tau} \tag{7.76}$$

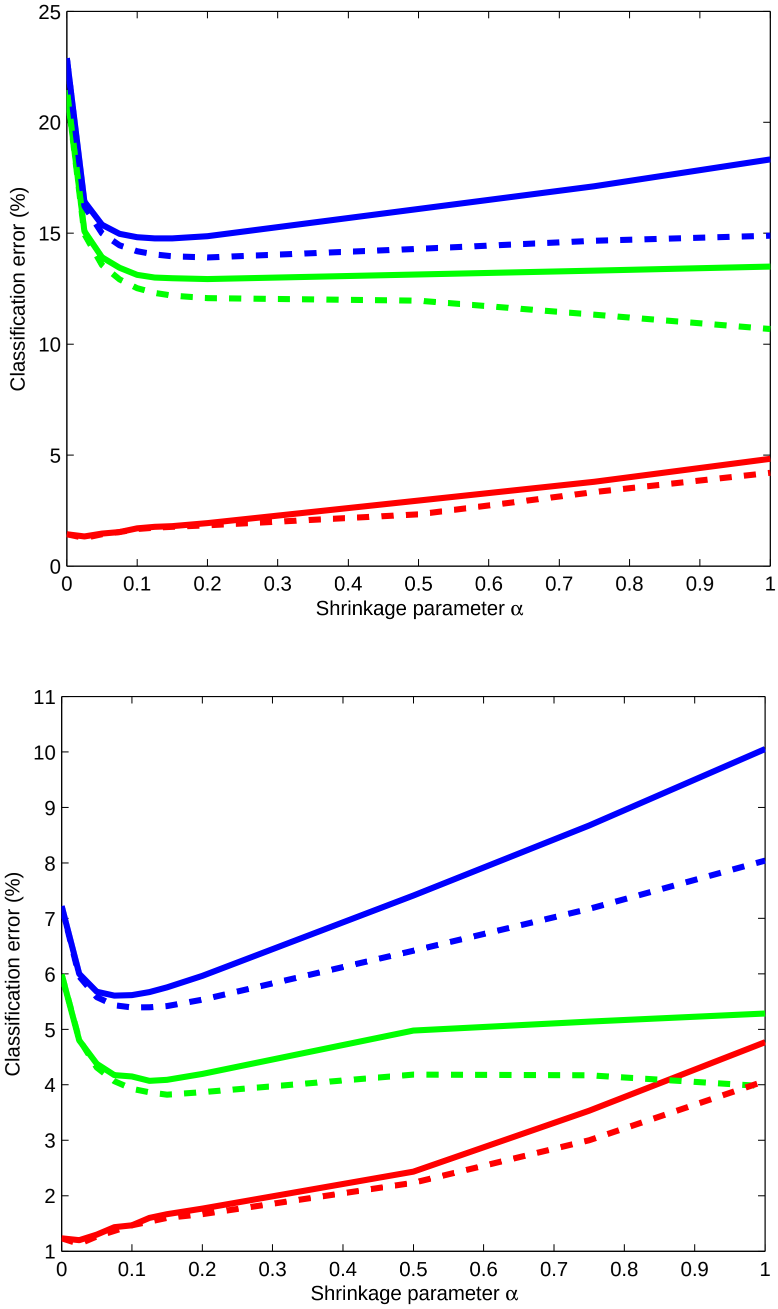

Figure 7.4: Error, variance and bias (as in Figure 7.2) for full covariance shrinkage models, with varying shrinkage parameter $\alpha$, using a standard ML-trained diagonal prior (solid) and an MMI-trained prior (dashed). Models trained with three samples per class (top) and five samples per class (bottom)

We can therefore convert the shrinkage parameter $\alpha$ to an equivalent weight $\tau$ using $\tau = \alpha\beta_m(1-\alpha)^{-1}$. Then using the notation of Section 7.2.2, the smoothed parameters are obtained via the smoothed statistics

$$\hat{\mu}_m = \frac{x_m + \tau\mu_m^D}{\beta_m + \tau} \tag{7.77}$$

$$\hat{\Sigma}_m = \frac{Y_m + \tau(\Sigma_m^D + \mu_m^D\mu_m^{D^T})}{\beta_m + \tau} - \hat{\mu}_m\hat{\mu}_m^T \tag{7.78}$$

We use only ML statistics here; only the diagonal models are discriminatively trained. The dashed lines in Figure 7.3 show the performance with the MMI priors. A slightly lower minimum error rate is achieved. However, a lower error rate is maintained for a large range of values of $\alpha$ away from the optimum, making a good choice of $\alpha$ (or $\tau$) less important.

## 7.5 Summary

In this chapter we described various discriminative training criteria, having theoretical benefits for classification performance when the generative model used is not correct. We motivated the MCE and MMI criteria from a margin perspective. We discussed a standard approach to model training using the MMI criterion, focusing particularly on full covariance estimation in this setting. We discussed methods for improving the generalisation ability of discriminative estimators. Finally, we returned to the theme of Chapter 5: we presented simulations to illustrate the concepts of variance and bias in a discriminative setting, and, considering the use of full covariance models, motivated extensions of the use of a diagonal smoothing prior when discriminative training criteria are used.

# Chapter 8

# Discriminative training experiments

In this chapter we investigate incorporating discriminative training techniques into full covariance model estimation. We present results on the conversational telephone speech recognition task used for the earlier experiments in Section 6.2. Our objective is to obtain an optimal recipe for full covariance training.

## 8.1 Diagonal baseline systems

Our baseline diagonal system was the same as that used for the earlier experiments: a global HLDA transform was used to project the feature vector from 52 to 39 dimensions. Speaker adaptation was performed on the test set using block-diagonal CMLLR transforms with 32 regression classes per speaker. All transforms were estimated using the ML criterion. Our early experiments were performed without SAT; however we later repeated a selection of the experiments with SAT. When using SAT, CMLLR transforms were estimated for the training speakers using the baseline HLDA models, and these transforms were used for all subsequent SAT experiments. Recognition of the test utterances was carried out by rescoring baseline lattices with the new acoustic models, and then rescoring with the trigram language model to obtain a one-best transcription.

Following the procedure described in Young *et al.* (2006), numerator and denominator lattices were generated for each training utterance using the baseline models; in the denominator case, a bigram language model was used, and the lattices were heavily pruned. HDecode was used to add phone alignments to each arc. The lattices were rescored with a unigram language model, and the acoustic probabilities were scaled by 1/12, the inverse of the language model scaling factor used during decoding. The same lattices were used for all discriminative training.

The mean and variance parameters of the ML-trained diagonal models were updated using MMI training. We adopted the standard method (Povey & Woodland, 2000) of flooring the smoothing constant $D_m$ at $2\beta_m^{\text{den}}$. We found that the performance reached a peak after around four EBW iterations. WER results for the MMI-trained diagonal systems are shown in Table 8.1. It can be seen that substantial performance gains over the ML-trained models are achieved. All gains obtained using MMI are statistically significant; in addition, the use CMLLR and SAT continued to yield significant improvements with MMI.

| System | ML | MMI (by iteration) | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| HLDA | 35.5 | 34.2 | 33.5 | 33.1 | 33.1 |
| HLDA + CMLLR | 33.1 | 32.4 | 31.8 | 31.4 | 31.2 |
| HLDA + SAT + CMLLR | 32.9 | 32.0 | 31.3 | 31.1 | 30.9 |

Table 8.1: %WER for diagonal covariance systems with means and variances updated with MMI training.

## 8.2 Discriminatively trained priors

The experiments in Chapter 6 showed that the performance of full covariance models is dramatically improved when a diagonal smoothing prior is used. In Section 7.4.3 we proposed substituting the ML-trained prior for a discriminatively trained prior, as a means of the reducing the increase in bias when backing off to the diagonal models; the potential advantages were demonstrated by simulations.

We applied the technique to the CTS models. To ensure a match between the full covariance statistics for each Gaussian and the respective priors used, the means and Gaussian weights were fixed to the same values for both. In both cases, the values used were those obtained by four iterations of MMI training (the models whose performance was reported in the previous section). Full covariance statistics were accumulated, centred about the fixed means, using the standard ML state posteriors. These were smoothed with the diagonal prior according to the formula (7.78). Full covariance discriminative training was not performed at this stage, but was applied in later experiments described in Section 8.3. Speaker adaptation was performed using the CMLLR transforms estimated for the initialising models.

Table 8.2 compares the effects of three different smoothed full covariance systems: the first column shows the standard ML-trained smoothed full covariance systems, used in the experiments in Chapter 6. The second shows full covariance models with MMI-trained means, smoothed with an equivalent ML-trained diagonal prior; the third shows the full covariance models smoothed with MMI-trained smoothed with the MMI-trained diagonal covariance models. In each case we investigated the performance for a range of values of $\tau$, the prior weight. We also estimated models using the prior weight obtained from the optimal shrinkage parameter $\alpha$ using global sharing of the shrinkage statistics (this is labelled as "shrinkage"). We did not adjust the formulae for estimating $\alpha$ to make them explicitly discriminative. The results are shown graphically in Figure 8.1.

The results show that when the diagonal models are discriminatively trained, the use of full covariance models results in a smaller performance improvement over the diagonal models – in fact, the differences are not statistically significant. This is to be expected, since discriminative training compensates for the lack of model-correctness, which is of greater importance when there are fewer parameters. However, the necessity of off-diagonal smoothing is again demonstrated, yielding significant improvements over both diagonal and unsmoothed full covariance systems. Furthermore, the use of the discriminatively-trained diagonal prior leads to a lower WER minimum, and also results in a low WER being maintained for a wide range of values of $\tau$. The simulations carried out Section 7.4.3 appear to mirror the performance of the ASR models fairly well. The method for analytically-obtaining the optimal prior weight continues to yield results close to the optimum for all three systems, again, with the differences not significant.

The experiments using the MMI prior were repeated with SAT systems. The results are given in Table 8.3 and Figure 8.2. Again, the new systems are compared to the original ML systems. The results show the same trends as for the

| | Initialising model (prior model) | | |
|---|---|---|---|
| Prior $\tau$ | ML (ML) [a] | MMI (ML) [b] | MMI (MMI) [c] |
| 0 | 32.1 | 31.5 | 31.1 |
| 10 | 31.3 | 30.7 | 30.7 |
| 20 | 31.0 | 30.5 | 30.5 |
| 40 | 30.7 | 30.3 | 30.2 |
| 60 | 30.6 | 30.3 | 30.1 |
| 80 | 30.5 | 30.3 | 30.1 |
| 100 | 30.5 | 30.3 | 30.1 |
| 125 | 30.6 | 30.4 | 30.0 |
| 150 | 30.7 | 30.5 | 30.1 |
| 175 | 30.7 | 30.5 | 30.1 |
| 200 | 30.8 | 30.6 | 30.1 |
| 300 | 31.0 | 30.8 | 30.1 |
| 400 | 31.3 | 30.9 | 30.2 |
| Diagonal | 33.3 | - | 31.2 |
| Shrinkage | 30.6 | 30.4 | 30.1 |

Table 8.2: %WER results for full covariance systems, with covariance paramters updated with ML and smoothed with a diagonal prior. Models use HLDA and CMLLR.

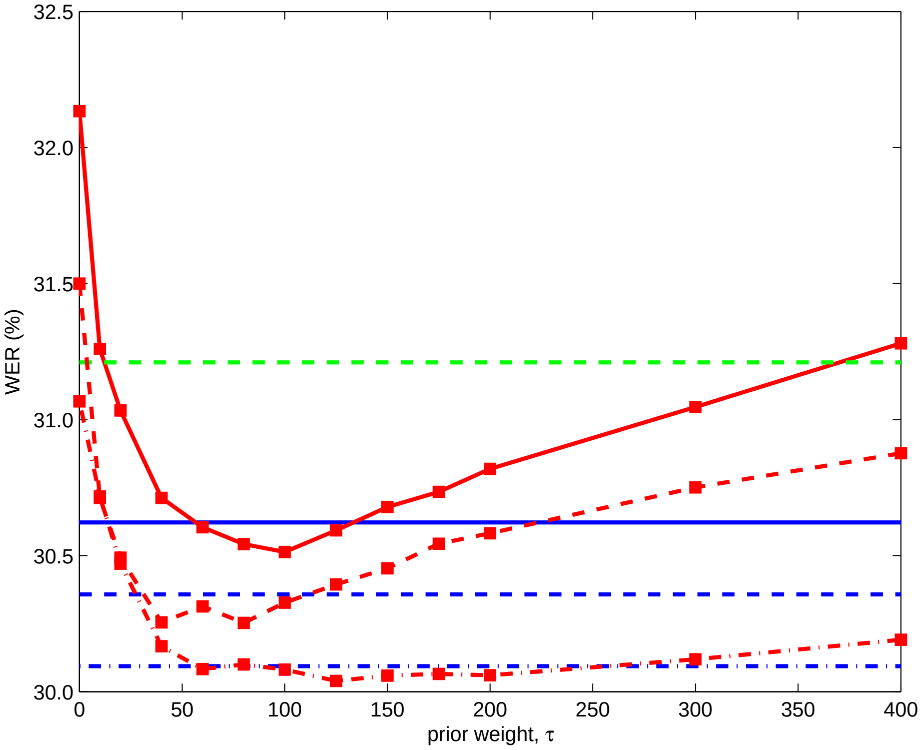

Figure 8.1: %WER results for the systems referred to in Table 8.2, shown as [a] solid, [b] dashed and [c] dash-dotted. In each case, blue lines show the shrinkage estimator; the green dashed line shows the diagonal MMI-trained models (diagonal ML-trained models are not shown for reasons of plot scaling).

non-SAT systems. The addition of SAT again resulted in statistically significant performance improvements for the optimal smoothed systems.

| | Prior/initialising model | |
|---|---|---|
| Prior $\tau$ | ML [a] | MMI [c] |
| 0 | 32.1 | 31.0 |
| 10 | 31.1 | 30.6 |
| 20 | 30.9 | 30.3 |
| 40 | 30.5 | 30.1 |
| 60 | 30.4 | 29.9 |
| 80 | 30.3 | 29.9 |
| 100 | 30.4 | 29.9 |
| 125 | 30.4 | 29.9 |
| 150 | 30.4 | 29.9 |
| 175 | 30.5 | 29.9 |
| 200 | 30.6 | 29.9 |
| 300 | 30.9 | 30.0 |
| 400 | 31.1 | 30.0 |
| Diagonal | 32.9 | 30.9 |
| Shrinkage | 30.4 | 29.9 |

Table 8.3: %WER results for SAT-trained full covariance systems, with covariance parameters updated with ML and smoothed with a diagonal prior.

## 8.3 Discriminative full covariance updates

Finally, we performed MMI estimation directly on the full covariance models. It is essential to initialise the off-diagonal elements with ML estimation before applying MMI updates. Since full covariance models are prone to rapid over-training, the choice of this initial full covariance model is important. A related issue is the question of how off-diagonal smoothing, shown to bring benefits to ML-trained models, may be incorporated; and whether it continues to yield advantages when full covariance MMI training is applied.

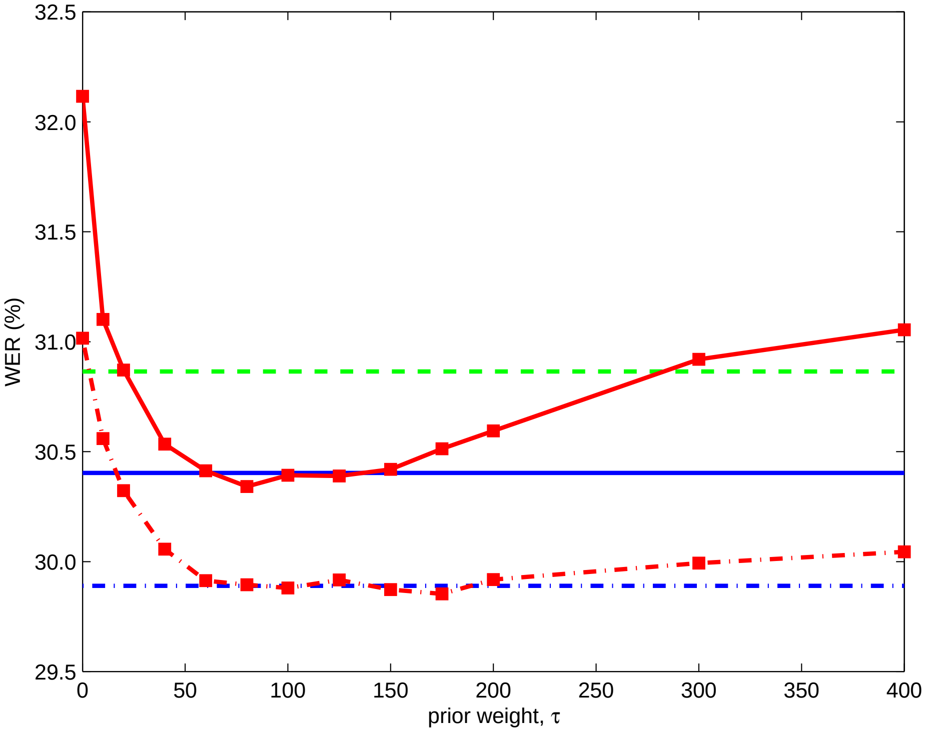

Figure 8.2: %WER results for the systems referred to in Table 8.3, shown as [a] solid and [c] dash-dotted. In both cases, blue lines show the shrinkage estimator; the green dashed line shows the diagonal MMI-trained models (diagonal ML-trained models are not shown for reasons of plot scaling).

If a diagonal prior is used for the ML estimation used generate the initialising models, then it influences the MMI estimation via the smoothing term $G^{\mathrm{sm}}(\theta, \theta^0)$. (In addition, the state posteriors used in the EBW algorithm are likely to be more accurate). Under the standard method for setting the smoothing constant $D_m$, the influence of the initial model is likely to be greater in the full covariance case that the diagonal covariance case due to the higher dispersion of the eigenvalues of (7.50) in the former case. An alternative is to apply the prior in the style of MMI-MAP (Povey *et al.*, 2003). Here the diagonal prior is used to obtain a MAP (shrinkage) estimate:

$$\mu_m^{\mathrm{map}} = \frac{x_m + \tau\mu_m^D}{\beta_m + \tau} \tag{8.1}$$

$$\Sigma_m^{\mathrm{map}} = \frac{Y_m + \tau(\Sigma_m^D + \mu_m^D \mu_m^{D\,T})}{\beta_m + \tau} - \mu_m^{\mathrm{map}} \mu_m^{\mathrm{map}\,T} \tag{8.2}$$

which in turn is used as a prior for the MMI estimation, leading to parameter updates given by:

$$\hat{\mu}_m = \frac{x_m^c + x_m^{sm} + \tau^{\mathrm{map}}\mu_m^{\mathrm{map}}}{\beta_m^c + \beta_m^{sm} + \tau^{\mathrm{map}}} \tag{8.3}$$

$$\hat{\Sigma}_m = \frac{Y_m^c + Y_m^{sm} + \tau^{\mathrm{map}}(\Sigma_m^{\mathrm{map}} + \mu_m^{\mathrm{map}}\mu_m^{\mathrm{map}\,T})}{\beta_m^c + \beta_m^{sm} + \tau^{\mathrm{map}}} - \hat{\mu}_m \hat{\mu}_m^T \tag{8.4}$$

where $\tau^{\mathrm{map}}$ is a new prior constant, which Povey *et al.* (2003) suggest setting to 100.

We performed full covariance MMI training with a range of initialising full covariance systems, updating only covariance matrices. Table 8.4 compares WER results across these systems (described using the notation from Tables 8.2 and 8.3). In each case, results are shown from the original diagonal system are presented; then the full covariance system obtained from that diagonal system using ML estimation; and finally, the results following an iteration of MMI estimation. Initial experiments were performed without SAT; we later repeated the most successful training recipes with SAT. The smoothed systems shown in the table use a prior weight $\tau = 100$.

As with the original ML estimation, a single full covariance MMI update was performed: we found that further iterations reduced performance. Applying an additional prior resulted in slight performance degradation when the initialising

| System | Diagonal | Initial | MMI |
|---|---|---|---|
| Systems without SAT | | | |
| [a] + naive FC | 33.3 | 32.1 | 31.1 |
| [a] + smoothed FC | 33.3 | 30.5 | 30.4 |
| [c] + naive FC | 31.2 | 31.1 | 30.2 |
| [c] + smoothed FC | 31.2 | 30.1 | 29.4 |
| [c] + shrinkage FC | 31.2 | 30.1 | 29.5 |
| Systems with SAT | | | |
| [c] + naive FC | 30.9 | 31.0 | 30.0 |
| [c] + smoothed FC | 30.9 | 29.9 | **29.2** |

Table 8.4: %WER results for MMI-trained full covariance systems, shown in the final column. The first column shows the WER for the original diagonal covariance system; the second shows the WER for the initialising full covariance system.

full covariance system had been estimated with smoothing. The results show that the performance of full covariance MMI estimation is indeed sensitive to the initial full covariance model: the need for off-diagonal smoothing is not removed by applying discriminative training. The differences between smoothed and unsmoothed systems are statistically significant in each case, as are the differences with and without the application of MMI training, except in the second row of the table where there was no MMI training of the diagonal models.

## 8.4 Summary

In this chapter we presented results on the Conversation Telephone Speech recognition task, incorporating discriminative training techniques into full covariance model training. Experiments showed that off-diagonal smoothing is essential to improve the performance of full covariance models over discriminatively-trained diagonal models; in addition, the use of a discriminatively trained prior (using the MMI criterion) gives improved performance over a generative prior, even when the full covariance parameters are trained using maximum likelihood, and the results are less sensitive to the choice of smoothing parameter. Updating the full covariance parameters using MMI yields further performance improvements, but this does not remove the need for off-diagonal smoothing at an earlier stage in the training process.

Combining all the results in this chapter suggests the following training procedure for full covariance models:

1. Train diagonal covariance models with the desired number of Gaussians using the EM algorithm.
2. Update the diagonal models using several iterations of discriminative training.
3. Train full covariance models using ML estimation, smoothing with the MMI-trained diagonal models.
4. Apply a further iteration of discriminative training to the full covariance models

# Chapter 9

# Conclusion

## 9.1 Work done

When using Gaussian mixture models as acoustic models for automatic speech recognition, effective modelling of the covariance matrices is important for good recognition performance on unseen speech data. This thesis investigated full covariance modelling for the Gaussian covariance matrices. We were motivated by previous work for ASR using the formalism of graphical modelling to specify a rich dependency structure between acoustic features, and first tried to improve upon methods for learning the model structure automatically from data.

We considered the graphical modelling problem in the context of estimating sparse precision matrices for Gaussian mixture models. We used recent results from outside the body of ASR literature to obtain efficient algorithms for simultaneous parameter and model structure learning using $l_1$-penalised maximum likelihood estimation. We implemented these techniques to estimate the covariance parameters of an HMM-GMM system for ASR.

The early graphical modelling work, particularly the benefits of the bounds on matrix conditioning imposed by the penalised maximum likelihood estimation, prompted the main research question of the thesis:

> *Full covariance models are capable of higher modelling power than alternatives, but how can we overcome the difficulties in parameter estimation when training data is limited?*

We identified three requirements for effective full covariance estimation:

- covariance matrices should be well-conditioned;
- full covariance models should generalise well to unseen test data, even when trained on limited data;
- parameter estimation should compensate for the fact that underlying modelling assumptions are not correct.

These considerations led us to investigate the use of a shrinkage estimator as an alternative full covariance estimator to the standard sample covariance matrix, which can be viewed as a method of off-diagonal smoothing.

We discussed the beneficial properties of the shrinkage estimator with regard to the requirements of matrix conditioning and generalisation. We derived formulae for the estimation of the optimal shrinkage parameters in a GMM system, and obtained a method for sharing the relevant statistics across multiple Gaussians in a system. We related the shrinkage techniques to a Bayesian approach to full covariance estimation using a diagonal smoothing prior.

When applying statistical models for classification, the invalidity of model correctness assumptions gives the need for discriminative training to be applied. We therefore considered the properties of the shrinkage estimator in a discriminative context. We integrated the smoothing techniques into discriminative parameter estimation with MMI estimation, developing recipes for full covariance training.

## 9.2 Outcomes

To test the various techniques, we carried out initial experiments on the TIMIT phone classification task, varying the quantity of training data to simulate sparse-data conditions. We found the graphical modelling approach did not yield improvements over the better-performing of the standard full covariance and diagonal covariance models. However, the shrinkage estimator was found to yield consistent performance improvements over both diagonal covariance and standard full covariance systems, regardless of the quantities of training data used. However, the best results could almost always obtained using a single, hand-tuned prior weight.

We evaluated the smoothed full covariance systems on a large vocabulary conversational telephone speech recognition task. On this task, we found that off-diagonal smoothing is essential for the good performance of full covariance models. With the best smoothing weight, the performance improvement over diagonal-covariance systems was more than doubled compared to the standard full covariance models; the results were similarly improved when the optimal shrinkage parameter was estimated from data. With speaker adaptation applied during training and decoding, word error rate was reduced from 32.9% with diagonal models to 30.3% with the best smoothed system. In contrast, a standard full covariance system achieved a rate of 32.1%.

When discriminative training was applied, the performance of diagonal covariance models was reduced to 30.9%. The best system without smoothing achieved a rate of 30.0%. We demonstated that the application of the smoothing technique continued to give significant improvements to the discriminatively-trained system, leading to a reduction in WER to 29.2%.

## 9.3 Future work

We briefly consider how some of the shortcomings of the work presented in this thesis might be addressed in future work, and discuss directions for related research. Both sections are somewhat speculative.

Given the additional statistics obtained from the data, it is somewhat frustrating that is not possible to set the shrinkage parameter on a Gaussian or class-specific basis such that the models outperform those with any global smoothing factor. This would presumably require some form of clustering but it is not clear how this could be best performed. Alternatively, it is possible that the generative approach to shrinkage parameter estimation is not suitable when the goal is accurate classification. We discuss this further in Section 9.3.1 below.

An application of full covariance modelling that we have we have considered only briefly in this work is in the estimation of full-covariance linear transforms for speaker adaptation, most notably CMLLR. Here, limited-data techniques may often be more important due to the lack of adaptation data. We discuss this briefly in Section 9.3.2.

### 9.3.1 Shrinkage estimators with an MMI-based performance measure

In Section 7.4 we presented simulations of a bias-variance decomposition using a loss function matching the MCE criterion for training. The results suggested that the optimal shrinkage parameter for the goal of good classification performance is not the same as the optimal parameter from a generative modelling perspective. We would like to be able to be able to optimise the parameter with regard to a discriminative criterion.

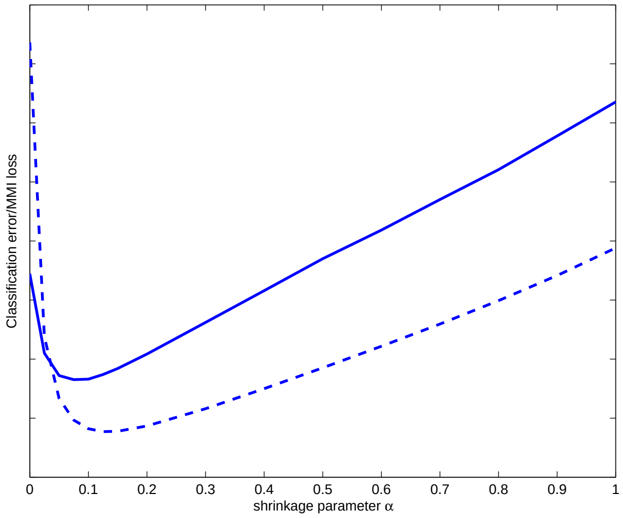

Figure 9.1: Comparing the MCE (solid) and MMI (dashed) loss functions for shrinkage estimators with varying $\alpha$, 5 samples per class. The loss functions have been scaled so that the minima of both may be compared on a single plot.

It is possible to replace the MCE loss function by an MMI-like expected loss:

$$\mathbb{E}_{p(x,y)}[\mathbb{E}_{\mathcal{T}_R} \log p_{\theta(\mathcal{T}_R)}(y|x)] \tag{9.1}$$

To simulate the performance of the models using this loss, we repeat the procedure of sampling large numbers of small test sets to estimate the inner expectation, then estimating the outer expectation on a large test set. The MMI loss is, however, highly sensitive to outlier observations in the test set (those with very low

$p_\theta(x_r|y_r)$ under the ML model) with which lead to very low posterior probabilities even if the models are well-trained. To avoid this, we use a similar approach to Sha & Saul (2007), and remove the lowest 10% of outliers.

As an example, Figure 9.1 compares this MMI loss function with the MCE loss, for varying $\alpha$. The minima with respect to $\alpha$ are close. Given that we can consistently estimate the classical variance by using additional sample statistics, we speculate that it may be possible to derive analytic methods to estimate the minimum of the MMI loss function without need for cross-validation, based on these statistics. Further work is needed here.

### 9.3.2 Applications to speaker adaptation

We speculate that the full covariance techniques presented here may find application to the common approach to speaker adaptation using full-covariance linear transforms. Recall from (2.57) in Section 2.4.2 that to perform CMLLR adaptation, speaker-specific full covariance statistics $S_m$ are required, given by

$$S_m = \frac{\sum_t \gamma_m(t)(\zeta_t - \bar{\zeta}_m)(\zeta_t - \bar{\zeta}_m)^T}{\sum_t \gamma_m(t)} \tag{9.2}$$

These are aggregated over all Gaussians in the adaptation class. Estimates of the transform parameters, based on these statistics, may be unreliable when the amount of adaptation data for a given speaker is limited.

The standard remedy to the problem of limited data is to use tying to reduce the number of adaptation classes, effectively increasing the amount of data available for estimating each transform; other implementations of CMLLR often may ensure more reliable estimation by restricting the transforms to be diagonal or block-diagonal. However, these solutions reduce modelling power and hence discriminative ability. We highlight two recently-proposed alternatives: (Ghoshal *et al.*, 2010; Povey *et al.*, 2010) have presented a subspace technique where the speaker transform $R^{(s)}$ is decomposed the linear combination of $B$ basis matrices $W_b$:

$$R^{(s)} = W_0 + \sum_b^B \lambda_b^{(s)} W_b \tag{9.3}$$

and also provide a method for optimising the coefficients $\lambda_b$. This has the advantage of reducing the number of parameters that must be estimated whilst retaining the full-covariance form of the transforms.

Yamigishi *et al.* (2009) developed Constrained Structural MAP Linear Regression (CSMAPLR). Here, CMLLR is employed, but the statistics used are smoothed with lower-dimensional priors using a MAP approach, explicitly maintaining the robustness of the estimation when data is limited. In this work, the authors applied the technique only to speaker adaptation of HMM-based models for text-to-speech. To our knowledge, results using the same technique for ASR have not yet been reported.

These respective approaches are somewhat analogous to covariance modelling using subspace methods and using smoothed full covariance models. It is the latter approach that suggests a use of the shrinkage techniques presented in this thesis, for adaptation within a similar framework to CSMAPLR, where improvements might be gained with appropriate choices of prior and smoothing parameters.

# Bibliography

ANASTASAKOS, T., MCDONOUGH, J., SCHWARTZ, R. & MAKHOUL, J. (1996). A compact model for speaker adaptive training. In *Proc. ICSLP*. 2.4.3

AXELROD, S., GOEL, V., GOPINATH, R.A., OLSEN, P.A. & VISWESWARIAH, K. (2005). Subspace constrained Gaussian mixture models for speech recognition. *IEEE Transactions on Speech and Audio Processing*, **13**, 1144–1160. 3.3.3, 3.4, 3.5, 6.1.2, 6.1.4

BAHL, L., JELINEK, F. & MERCER, R. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Journal of Pattern Analysis and Machine Intelligence*, **5**, 179–190. 1.1

BAHL, L., BROWN, P., DE SOUZA, P. & MERCER, R. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc ICASSP*, vol. 2, 613–616. 1.2, 3.1.1, 7.1.2

BAKER, J. (1975). The Dragon system – an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **13**, 24–29. 1.1

BANERJEE, O., D'ASPREMONT, A. & GHAOUI, L.E. (2006). Convex optimization techniques for fitting sparse gaussian graphical models. In *Proc. ICML*, Pittsburgh, PA. 1.2, 4.4.1, 4.4.2, 4.4.2, 1, 4.4.3, 4.4.4, A.3, A.3.9

BAUM, L., PETRIE, T., SOULES, G. & WEISS, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic function of markov chains. *Annals of Mathematical Statistics*, **41**, 164–171. 2.1.3, 2.3.2

BELL, P. & KING, S. (2008). Covariance updates for discriminative training by constrained line search. In *Proc. Interspeech*. 7.2.2

BILMES, J. (1998). Maximum mutual information based reduction strategies for cross-correlation based joint distributional modeling. In *Proc. ICASSP*. 4.2.1

BILMES, J. (1999). Buried Markov models for speech recognition. In *Proc. ICASSP*, Phoenix. 1.2, 4, 4.1.1

BILMES, J. (2000a). Dynamic Bayesian multinets. In *Proc. 16th Conference on Uncertainty in Artificial Intelligence*, Stanford. 4.1.1, 4.2.1

BILMES, J. (2000b). Factored sparse inverse covariance matrices. In *Proc. ICASSP*, Istanbul, Turkey. 4.2.1, 6.1.3

BILMES, J. & BARTELS, C. (2005). Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, **22**, 89–100. 4.1.1

BILMES, J., RICHARDSON, T., FILALI, K., LIVESCU, K., XU, P., JACKSON, K., BRANDMAN, Y., SANDNESS, E., HOLTZ, E., TORRES, J. & BYRNE, B. (2001). Discriminatively structured graphical models for speech recognition. CSLP 2001 summer workshop final report, Johns Hopkins University. 4.2.1

BISHOP, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. 1.1

BOUCHARD, G. & TRIGGS, B. (2004). The trade-off between generative and disciminative classifiers. In J. Antoch, ed., *Proc. COMPSTAT*, Physica-Verlag. 7.1.2, 7.3.2

BOURLARD, H. & MORGAN, N. (1994). *Continuous Speech Recognition: a Hybrid Approach*. Kluwer Academic Publishers. 7.1

BOYD, S. & VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge University Press. A.3

BURGES, C.J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167. 7.3.3

CHEN, S. & GOODMAN, J. (1999). An empirical study of smoothing techniques for language modelling. *Computer Speech and Language*, **13**, 359–394. 2.2.3

CHEN, S., KINGSBURY, B., MANGU, L., POVEY, D., SAON, G., SOLTAU, H. & ZWEIG, G. (2006). Advances in speech transcription at IBM under the darpa ears program. *IEEE Transactions on Audio, Speech and Language processing*, **14**, 1596–1608. 3.4, 5, 5.1.4, 7.2.3

DAHL, J., VANDENBERGHE, L. & ROYCHOWDHURY, V. (year unknown). Covariance selection for non-chordal graphs via chordal embedding, unpublished article. 4.3.1

DAVIS, S. & MERMELSTEIN, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-28**, 355–366. 2.2.1

DEMPSTER, A. (1972). Covariance selection. *Biometrics*, **28**, 157–175. 1.2, 4.1.3, 4.3.1, 4.3.1, 4.4.1

DEMPSTER, A., LAIRD, N. & RUBIN, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38. 2.3.2

DIGALAKIS, V.V., RTISCHEV, D. & NEUMEYER, L. (1995). Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, **3**. 2.4.2

DOMINGOS, P. (2000). A unified bias-variance decomposition for zero-one and squared loss. In *Proc. AAAI*, 231–238, Morgan Kaufmann. 7.4.2, 7.4.2, 7.4.2

DOUMPIOTIS, V. & BYRNE, W. (2004). Pinched lattice minimum Bayes risk discriminative training for large vocabulary continuous speech recognition. In *Proc. Interspeech*. 7.1.3

DUDA, R. & HART, P. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York. 2.4.1

ELLIS, D.P. & BILMES, J.A. (2000). Using mutual information to design feature combinations. In *Proc. ICSLP*, Beijing. 4.2.1

FERGUSON, J.D., ed. (1980). *Hidden Markov Models for Speech*. Institute for Defense Analyses, Princeton. 1.1

FITT, S. (2000). Documentation and user guide to Unisyn lexicon and post-lexical rules. Tech. rep., Centre for Speech Technology Research, University of Edinburgh. 6.2.1

GALES, M. (1997). Adapting semi-tied full-covariance HMMs. Tech. rep., Cambridge University Engineering Department. 2.4.1, 6.2.3

GALES, M. (1998). Maximum likelihood linear transforms for HMM-based speech recognition. *Computer Speech and Language*, **12**, 75–98. 2.4.2, 2.4.2, 2.4.3

GALES, M. (1999). Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, **7**, 272–281. 2.4.1, 3.3.2, 3.3.2, 6.1.2

GALES, M. & WOODLAND, P. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, **10**, 249–264. 2.4.2

GARAU, G. (2008). *Speaker Normalisation for Large Vocabulary Multiparty Conversational Telephone Speech Recognition*. Ph.D. thesis, University of Edinburgh. 6.2.1

GAUVAIN, J.L. & LEE, C.H. (1994). Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, **2**. 2.4.2

GHOSHAL, A., POVEY, D., AGARWAL, M., AKYAZI, P., BURGET, L., KAI, F., GLEMBEK, O., GOEL, N., KARÁFIAT, M., RASTROW, A., ROSE, R., SCHWARZ, P. & THOMAS, S. (2010). A novel estimation of feature-space MLLR for full-covariance models. In *Proc. ICASSP*. 9.3.2

GOPALAKRISHNAN, P., KANEVSKY, D., NADAS, A. & NAHAMOO, M., D.AND PICHENY (1988). Decoder selection based on cross-entropies. In *Proc. ICASSP*. 7.3.2

GOPINATH, R., RAMABHADRAN, B. & DHARANIPRAGADA, S. (1998). Factor analysis invariant to linear transformations of data. In *Proc. ICSLP*. 3.3.4

Grimmet, G. & Stirzaker, D. (1982). *Probability and Random Processes*. Oxford University Press. 2.1.1

Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., McCowan, I., Moore, D., Wan, V., Ordelman, R. & Renals, S. (2005a). The 2005 AMI system for the transcription of speech in meetings. In *Proceedings of the Rich Transcription 2005 Spring Meeting Recognition Evaluation*. 6.2.1

Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., McCowan, I., Moore, D., Wan, V., Ordelman, R. & Renals, S. (2005b). The development of the AMI system for the transcription of speech in meetings. In *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*. 6.2.1

Hain, T., Dines, J., Garau, G., Karafiat, M., Moore, D., Wan, V., Ordelman, R. & Renals, S. (2005c). Transcription of conference room meetings: an investigation. In *Proc. Interspeech*. 6.2.1

Hermansky, H. (1990). Perceptual linear predictive analysis of speech. *The Journal of the Acoustical Society of America*, **87**, 1738–1752. 1.1, 2.2.1

Jiang, H., Li, X. & Liu, C. (2006). Large margin hidden Markov models for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, **14**, 1584–1595. 7.3.3

Jones, B. & West, M. (2005). Covariance decomposition in undirected graphical models. *Biometrika*, **92**, 779–786. 4.1.3

Juang, B.H. & Katagiri, S. (1992). Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, **40**, 3043–3053. 7.1.1

Kaiser, J., Horvat, B. & Kačič, Z. (2000). A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Proc. ICSLP*. 7.1.3

Kumar, N. & Andreou, A.G. (1998). Hetroschedastic discriminant analysis and reduced rank HMMs for improved recognition. *Speech Communication*, **26**, 283–297. 2.4.1

Lafferty, J., McCallum, A. & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *Proc ICML*. 7.1

Lauritzen, S.L. (1996). *Graphical Models*. Oxford University Press. 4, 4.1.3

Ledoit, O. & Wolf, M. (2003). Imporved estimation of the covariance matrix of stock returns with appication to portfolio selection. *Journal of Empirical Finance*, **10**. 5, 5.1.1

Ledoit, O. & Wolf, M. (2004). A well-conditioned estimator for large covariance matrices. *Journal of Multivariate Analysis*, **88**, 365–411. 5, 5.1.2, 5.1.3, 5.1.3, 5.2.1, 5.2.2, A.2

Lee, K. & Hon, H. (1988). Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**, 1641–1648. 6.1.1

Leggetter, C. & Woodland, P. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, **9**, 171–185. 2.4.2

Li, J., Yuan, M. & Lee, C.H. (2006). Soft margin estimation of hidden Markov model parameters. In *Proc. Interspeech*. 7.3.3

Li, X.W. & Jiang, H. (2006). Solving large margin HMM estimation via semidefinite programming. In *Proc. ICSLP*. 7.3.3

Liu, P. & Soong, F. (2008). An ellipsoid constrained quadratic programming perspective to discriminative training of HMMs. In *Proc. Interpseech*, Brisbane. 7.2.2

Liu, P., Liu, C., Jiang, H., Soong, F.K. & Wang, R.H. (2007). A constrained line search approach to general discriminative HMM training. In *Proc. ASRU*, Kyoto. 7.2.2

LIU, X. & GALES, M.J.F. (2007). Automatic model complexity control using marginalized discriminative growth functions. *IEEE Transactions on Audio, Speech and Processing*, **4**. 3.5

MARKEL, J. & GRAY, J., A.H. (1976). *Linear Prediction of Speech*. Springer-Verlag. 1.1

MCDERMOTT, E. & KATAGIRI, S. (2004). A derivation of minimum classification error from the theoretical classification risk using Parzen estimation. *Computer Speech and Language*, **8**, 107–122. 7.1.1

MEERBERGEN, K. & TISSEUR, F. (2001). The quadratic eigenvalue problem. *SIAM Review*, **43**, 235–286. A.1

MEINSHAUSEN, N. (2005). A note on the Lasso for Gaussian graphical model selection, unpublished work. 4.4.4

MEINSHAUSEN, N. & BÜHLMAN, P. (2006). High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, **34**, 1436–1462. 1.2, 4.4.1

NÁDAS, A. (1983). A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE transactions on Acoustics, Speech and Signal Processing*, **31**, 814–817. 7.1.2, 7.3.2

NEY, H., ESSEN, U. & KNESER, R. (1994). On structure probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, **8**, 1–38. 2.2.3, 6.2.1

NORMANDIN, Y. & MORGERA, S.D. (1991). An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. In *Proceedings ICASSP*, 537–540, Toronto. 3.1.1, 7.2.1, 7.2.2

OLSEN, P. & GOPINATH, R.A. (2004). Modeling inverse covariance matrices by basis expansion. *IEEE Transactions on Speech and Audio Processing*, **12**, 37–46. 3.3.3

PALLETT, D.S., FISHER, W.M. & FISCUS, J.G. (1990). Tools for the analysis of benchmark speech recognition tests. In *Proc. ICASSP*, vol. 1, 97–100. 6.2.1

PENNONI, F. (2004). Fitting directed graphical gaussian models with one hidden variable. *Development in Statistics, Metodoloski Zvezki, (Advances in Methodology and Statistics)*, **1**, 119–130. 4.1.2

PORTEOUS, B.T. (1985). *Properties of log-linear and covariance selection models*. Ph.D. thesis, University of Cambridge. 4.1.3

POVEY, D. (2003). *Discriminative Training for Large Vocabulary Speech Recognition*. Ph.D. thesis, Cambridge University Engineering Department. 5.1.4, 6.2.1, 7.2.2

POVEY, D. (2006). SPAM and full covariance for speech recognition. In *Proc. ICSLP*. 5.1.4

POVEY, D. (2009). Private communication. 6.1.2

POVEY, D. & SAON, G. (2006). Feature and model space speaker adaptation with full covariance gaussians. In *Proc. ICSLP*. 1.2, 2.4.2, 6.2.3

POVEY, D. & WOODLAND, P. (2000). Large-scale MMIE training for conversational telephone speech recognition. In *Proc. NIST Speech Transcription Workshop*, College Park, MD. 7.2.1, 7.2.2, 7.2.3, 8.1

POVEY, D. & WOODLAND, P. (2001). Improved discriminative training techniques for large vocabulary speech recognition. In *Proc. ICASSP*. 7.3.1

POVEY, D. & WOODLAND, P. (2002). Minimum phone error and I-smoothing for improved discrimative training. In *Proc. ICASSP*. 7.1.3, 7.3.2

POVEY, D., GALES, M., KIM, D. & WOODLAND, P. (2003). MMI-MAP and MPE-MAP for acoustic model adaptation. In *Proc. Eurospeech*. 8.3, 8.3

POVEY, D., KANEVSKY, D., KINGSBURY, B., RAMABHADRAN, B., SOAN, G. & VISWESWARIAH, K. (2008). Boosted MMI for model and feature-space discriminative training. In *Proc. ICASSP*. 7.3.2

POVEY, D., BURGET, L., AGARWAL, M., AKYAZI, P., KAI, F., GHOSHAL, A., GLEMBEK, O., GOEL, N., KARÁFIAT, M., RASTROW, A., ROSE, R., SCHWARZ, P. & THOMAS, S. (2010). Subspace Gaussian mixture models –

a structured model for speech recognition. Submitted to *Computer Speech and Language*. 9.3.2

Rabiner, L. & Juang, B.H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall. 2.1.3, 2.3.2

Rabiner, L.R., Juang, B.H., Levinson, S.E. & Sondhi, M.M. (1985). Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, **64**. 1.1

Richardson, F., Ostendorf, M. & Rohlicek, J. (1995). Lattice-based search strategies for large vocabulary recognition. In *Proc. ICASSP*, 576–579. 2.2.4

Rosti, A.V. & Gales, M.J.F. (2004). Factor-analysed hidden Markov models for speech recognition. *Computer Speech and Language*, **18**, 181–200. 3.3.4

Sakoe, H. & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-26**, 43–49. 1.1

Saon, G. & Povey, D. (2008). Penalty function maximization for large margin hmm training. In *Proc. Interspeech*. 7.3.3

Scanlon, P., Ellis, D.P.W. & Reilly, R. (2003). Using mutual information to design class-specific phone recognizers. In *Proc. Eurospeech*, 857–860, Geneva. 4.2.1

Schäfer, J. & Strimmer, K. (2005). A shrinkage approach to large-scale estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, **4**. 5, 5.1.3, 5.2.1, 5.2.1, 5.2.3

Schlüter, R. & Ney, H. (2001). Model-based MCE bound to the true Bayes' error. *IEEE Signal Processing Letters*, **8**. 7.1.2

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–464. 3.5

SHA, F. & SAUL, L.K. (2007). Large margin hidden Markov models for automatic speech recognition. *Advances in Neural Information Processing*, **19**, 1249–1256. 7.3.3, 9.3.1

SHIKANO, K. (1982). Spoken word recognition based upon vector quantization of input speech. *Trans. of Committee on Speech Research*, 473–480. 1.1

SIM, K. & GALES, M. (2004). Precision matrix modelling for large vocabulary continuous speech recognition. Tech. Rep. CUED/F-INFENG/TR.485, Cambridge University Engineering Department. 1.2

SIM, K. & GALES, M. (2005). Adaptation of precision matrix models on large vocabulary continuous speech recognition. In *Proc. ICASSP*. 6.2.3

SIM, K. & GALES, M. (2006). Minimum phone error training of precision matrix models. *IEEE Transactions on Speech and Audio Processing*, **14**, 882–889. 7.2.2, 7.2.3

SOLTAU, H., KINGSBURY, B., MANGU, L., POVEY, D., SAON, G. & ZWEIG, G. (2005). The IBM 2004 conversational telephony system for rich transcription. In *Proc. ICASSP*. 3.4

STEIN, C. (1956). Inadmissibility of the usual estimator of the mean of a multivariate normal distribution. In *Proc. Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 197–206. 1.2, 5.1.3

STEVENS, J., STANLEY SMITH; VOLKMAN & NEWMAN, E. (1937). A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, **8**, 185–190. 1.1

STOICA, P. & JANSSON, M. (2009). On maximum likelihood estimation in factor analysis – an algebraic derviation. *Signal Processing*, **89**, 1260–1262. 3.3.4

THOMPSON, B. (1990). Best-first enumeration of paths through a lattice – an active chart parsing solution. *Computer Speeech and Language*, **14**, 263–274. 2.2.4

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267–288. 4.4.1

VALENTINI, G. & DIETTERICH, T.G. (2004). Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *Journal of Machine Learning Research*, **5**, 725–775. 7.4.2, 7.4.2

VALTCHEV, V., ODELL, J., WOODLAND, P. & YOUNG, S. (1997). MMIE training of large vocabulary recognition systems. *Speech Communication*, **16**, 303–314. 6.1.2

VAPNIK, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag. 3.1.3, 3.1.3, 7.3.3

VINTSYUK, T. (1968). Speech discrimination by dynamic programming. *Kibernetika*. 1.1

YAMIGISHI, J., KOBAYASHI, T., NAKANO, Y., OGATA, K. & ISOGAI, J. (2009). Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. *IEEE Transactions on Audio, Speech and Language Processing*, **17**. 9.3.2

YOUNG, S. (2008). HMMs and related speech recognition technologies. In J. Benesty, M.M. Sondhi & Y. Huang, eds., *Springer Handbook of Speech Processing*, Springer. 2.2

YOUNG, S., ODELL, J. & WOODLAND, P. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proc. Human Language Technology Workshop*, Morgan Kaufman, San Francisco. 2.2.2

YOUNG, S., EVERMANN, G., GALES, M., HAIN, T., KERSHAW, D., LIU, X.A., MOORE, G., ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V. & WOODLAND, P. (2006). *The HTK Book*. Cambridge University Engineering Department. 2.2.1, 2.3.3, 8.1

YU, D., DENG, L., HE, X. & ACERO, A. (2008). Large-margin minimum classification error training: a theoretical risk minimization perspective. *Computer Speech and Language*, **22**, 415–429. 7.3.3

YUAN, M. & LIN, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35. 4.4.1

Zweig, G., Bilmes, J., Richardson, T., Filali, K., Livescu, K., Xu, P., Jackson, K., Brandman, Y., Sandness, E., Holtz, E., Torres, J. & Byrne, B. (2002). Structurally discriminative graphical models for automatic speech recognition: Results from the 2001 Johns Hopkins summer workshop. In *Proc. ICASSP*. 4.2.1

# Appendix A

# Derivations & Proofs

## A.1 The quadratic eigenvalue problem

We describe the solution of the quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)v = 0 \tag{A.1}$$

where $M$, $C$ and $K$ are all $d$-dimensional symmetric matrices and $x \in \mathbb{R}^d$. We summarise here from Meerbergen & Tisseur (2001). In general there are $2d$ eigenvalues $\lambda$. We define

$$u = \lambda v \tag{A.2}$$

and then substitute this into (A.1), to obtain

$$\lambda M u + C u + K v = 0 \tag{A.3}$$

The joint solution of (A.2) and (A.3) can be written as

$$\begin{pmatrix} -K & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} - \lambda \begin{pmatrix} C & M \\ I & 0 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} \tag{A.4}$$

or equivalently

$$\begin{pmatrix} -K & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} - \lambda \begin{pmatrix} C & M \\ M & 0 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} \tag{A.5}$$

This is a linear eigenvalue problem in $2d$ dimensions,

$$Ax - \lambda B x = 0 \tag{A.6}$$

where

$$A = \begin{pmatrix} -K & 0 \\ 0 & M \end{pmatrix}, \quad B = \begin{pmatrix} C & M \\ M & 0 \end{pmatrix}, \quad x = \begin{pmatrix} v \\ u \end{pmatrix} \tag{A.7}$$

with $A$ and $B$ both symmetric matrices. This is solved by finding the eigenvalues of the symmetric matrix $B^{-1}A$.

## A.2 Eigenvalues of the sample covariance matrix

Let $S$, the sample covariance matrix, be an unbiased estimator for a true covariance matrix, $\Sigma$. In this section we show that the eigenvalues $S$, are, on average, more dispersed than the eigenvalues of the true covariance matrix, $\Sigma$, even though $S$ is unbiased. We summarise from Ledoit & Wolf (2004). Denote the eigenvalues of $S$ by $l_i$, and their mean by $\bar{l}$. (We use roman letters here to emphasise that they are estimates from training data, in contrast to the eigenvalues of $\Sigma$, which are fixed but unknown). Their expected dispersion is given by

$$\mathcal{D}(S) = \mathbb{E} \sum_{i}^{d} (l_i - \bar{l})^2 \tag{A.8}$$

Since the trace of a matrix is given by the sum of its eigenvalues, the diagonal elements of $S$ also have mean $\bar{l}$. Their dispersion about this mean obeys the following inequality:

$$\begin{aligned} \sum_{i}^{d} (S_{ii} - \bar{l})^2 &\leq \sum_{i}^{d} (S_{ii} - \bar{l})^2 + \sum_{i}^{d} \sum_{j \neq i}^{d} S_{ij}^2 \\ &= \operatorname{tr}(S - \bar{l}I)^2 \end{aligned} \tag{A.9}$$

For any rotation $R$ (having $R^T R = I$), we have

$$\frac{1}{d} \operatorname{tr} R^T S R = \frac{1}{d} \operatorname{tr} S = \bar{l} \tag{A.10}$$

We now consider the dispersion of the diagonal elements of $R^T S R$ about their mean:

$$\operatorname{tr}(R^T S R - \bar{l}I)^2 = \operatorname{tr}[R^T (S - \bar{l}I) R]^2 \tag{A.11}$$

$$= \operatorname{tr}(S - \bar{l}I)^2 \tag{A.12}$$

If we set $R$ to be $G$, the matrix of eigenvectors of $S$, $R^TSR$ is the diagonal matrix consisting of the eigenvalues of $S$. So we have

$$\mathrm{tr}(S - \bar{l}I)^2 = \sum_i^d (l_i - \bar{l})^2 \tag{A.13}$$

Comparing to the result in Equation A.9, this implies that the eigenvalues are the *most dispersed* diagonal elements of $R^TSR$ for any rotation $R$.

Similarly, the dispersion of the eigenvalues of $\Sigma$ is given by

$$\mathcal{D}(\Sigma) = \sum_i^d (\lambda_i - \bar{\lambda})^2 \tag{A.14}$$

If $\Gamma$ is the matrix of eigenvectors of $\Sigma$, then $\Gamma^T\Sigma\Gamma$ is the diagonal matrix consisting of the eigenvalues of $\lambda_i$ of $\Sigma$. Since $S$ is an unbiased estimator of $\Sigma$, we have $\mathbb{E}\bar{l} = \bar{\lambda}$, and also, $\Gamma^TS\Gamma$ is an unbiased estimator of $\Gamma^T\Sigma\Gamma$ ($\Gamma$ is a parameter, not a random variable). Therefore the dispersion of the diagonal elements of $\Gamma^T\Sigma\Gamma$ obeys:

$$\mathbb{E}\sum_i^d ([\Gamma^TS\Gamma]_{ii} - \bar{l})^2 \geq \sum_i^d ([\Gamma^T\Sigma\Gamma]_{ii} - \bar{l})^2 \tag{A.15}$$

$$= \sum_i^d (\lambda_i - \bar{\lambda})^2 = \mathcal{D}(\Sigma) \tag{A.16}$$

(The first step uses Jensen's inequality). However, from the earlier results we know that the most dispersed diagonal elements of $R^TSR$, for any rotation $R$, are found when $R = G$, the matrix of eigenvectors of $S$, and are given by the eigenvalues $l_i$. Therefore

$$\mathcal{D}(S) = \mathbb{E}\sum_i^d (l_i - \bar{l})^2 \tag{A.17}$$

$$= \mathbb{E}\sum_i^d ([G^TSG]_{ii} - \bar{l})^2 \tag{A.18}$$

$$\geq \mathbb{E}\sum_i^d ([\Gamma^TS\Gamma]_{ii} - \bar{l})^2 \geq \mathcal{D}(\Sigma) \tag{A.19}$$

So we expect the eigenvalues of $S$ to be more dispersed than the eigenvalues of $\Sigma$.

## A.3 Graphical model structure learning with penalised likelihood

In this appendix we provide further detail concerning the work of Banerjee *et al.* (2006) on the use of penalised likelihood maximisation for GM structure learning, using convex optimisation techniques. Unless noted, the theorems here are all long-established results. A useful reference is Boyd & Vandenberghe (2004). We have provided our own proofs of propositions A.3.5 and A.3.6.

### A.3.1 Norms and their duals

**Definition A.3.1.** If $\|.\|$ is a norm on $\mathbb{R}^k$, then the dual norm, denoted $\|.\|_*$ is defined by:

$$\|u\|_* = \sup_{\|x\| \leq 1} u^T x \tag{A.20}$$

**Proposition A.3.2.** *For any $x, u \in \mathbb{R}^k$ then*

$$x^T u \leq \|x\| \|u\|_* \tag{A.21}$$

*Proof.* For any $x$, define $\bar{x} = \frac{x}{\|x\|}$, so that $\|\bar{x}\| = 1$. Then for any $u$, by the definition

$$\begin{aligned} \|u\|_* &\geq \bar{x}^T u \\ \Rightarrow \|x\| \|u\|_* &\geq x^T u \end{aligned}$$

□

The converse of Proposition A.3.2 is also true: suppose we have two norms, $\|.\|_a$ and $\|.\|_b$. Then if for all $x$, $u$ we have $x^T u \leq \|x\|_a \|u\|_b$, and furthermore, if for every $u$, there exists an $x$ such that the relation holds with equality, then $\|.\|_{a*} = \|.\|_b$. In other words, $\|.\|_b$ is the dual of $\|.\|_a$. By the symmetry of the relation it we see also that $\|.\|_{b*} = \|.\|_a$ – the dual of a dual norm is the original norm.

**Example A.3.3.** Consider the entry-wise ($l_p$) norms $\|.\|_p$ and $\|.\|_q$, defined by $\|u\|_p = (\sum |u_i|^p)^{\frac{1}{p}}$, with $p, q \geq 1$. If

$$\frac{1}{p} + \frac{1}{q} = 1 \tag{A.22}$$

then the two norms are duals of each other.

This result can be obtained via Proposition A.3.2 using Hölder's Inequality, which itself is obtained from Young's Inequality for scalars $x, u$:

$$xu \leq \frac{1}{p}x^p + \frac{1}{q}u^q \tag{A.23}$$

where $p$ and $q$ obey the relation above. A special case is that $\|.\|_{1*} = \|.\|_\infty$ and vice verse. (This can also be verified directly from the definition of the dual norm). The $l_q$ matrix norms we use in this section, as in Chapter 4, are entry-wise norms. As a consequence these can be treated identically to norms on $\mathbb{R}^k$.

## A.3.2 The penalised likelihood problem

Recall from Equation 3.7 that to obtain a maximum likelihood estimate of the precision matrix, $P$, we can equivalently maximise

$$\log|P| - \operatorname{tr} PS \tag{A.24}$$

The aim is to maximise a penalised version of this expression, where the penalty term is some function of the parameters $P$, designed to encourage sparsity in the matrix, and hence a sparse graphical model structure.

It is of course desirable to ensure that the resulting problem is easy to solve. This can be best achieved by ensuring that it is convex: if the penalty term is simply a count of the number of non-zero parameters, for example, then the problem is not convex. A solution is to use a matrix norm of $P$ as the penalty term: converting the problem to a minimisation problem, the objective function is then

$$f(P) = \rho\|P\| + \operatorname{tr} PS - \log|P| \tag{A.25}$$

The penalty parameter, $\rho > 0$, can be used to control the size of the penalty, and hence (we shall see), in the special case where the $l_1$ norm is chosen, the sparsity of the solution. It can easily be verified that this function is convex, and that the constraint set, given by $P \succ 0$, is also convex.

### A.3.3 Solving the problem via the dual

We first find a linear lower bound for the objective function of the primal problem A.25. This can be achieved by writing the penalised covariance matrix by $U = S + \Theta$, introducing a dual variable $\Theta$.

**Proposition A.3.4.** *A lower bound for $f(P)$ on $P \succ 0$ is given by*

$$g(\Theta) = \underset{P \succ 0}{\mathit{inf}} \{\operatorname{tr}(S+\Theta)P - \log |P|\} \tag{A.26}$$

*provided that $\|\Theta\|_* \leq \rho$. (It is clear that this is linear in $\Theta$)*

*Proof.* As noted at the end of Section A.3.1, the dual of a dual norm is the original norm, so that $\|P\|$ is the dual of $\|\Theta\|_*$. Therefore

$$\|P\| = \sup_{\|\Theta\|_* \leq 1} \operatorname{tr} \Theta P \tag{A.27}$$

$$\Rightarrow \rho\|P\| = \sup_{\|\Theta\|_* \leq 1} \operatorname{tr} \rho\Theta P \tag{A.28}$$

$$= \sup_{\|\Theta\|_* \leq \rho} \operatorname{tr} \Theta P \tag{A.29}$$

Therefore for $\|\Theta\|_* \leq \rho$ we have

$$f(P) \geq \operatorname{tr} \Theta P + \operatorname{tr} SP - \log |P| \tag{A.30}$$

$$\geq \inf_{P \succ 0} \{\operatorname{tr}(S+\Theta)P - \log |P|\} = g(\Theta) \tag{A.31}$$

□

We can find the infimum by differentiating with respect to $P$ and setting the result equal to zero:

$$S + \Theta - P^{-1} = 0 \tag{A.32}$$

$$\Rightarrow P = (S+\Theta)^{-1} \tag{A.33}$$

with the condition $P \succ 0$ leading to the condition $(S+\Theta) \succ 0$. Substituting this into the expression for $g(\Theta)$ gives

$$g(\Theta) = \operatorname{tr} I - \log |S+\Theta|^{-1} = k + \log |S+\Theta| \tag{A.34}$$

where $k$ is the dimension of the matrix. The solution to the dual problem is found by maximising this expression, subject to the constraints $(S+\Theta) \succ 0$, $\|\Theta\|_* \leq \rho$; or equivalently, maximising $k + \log |U|$, subject to $\|U - S\|_* \leq \rho$.

### A.3.4 Properties of the solution

We denote an optimal point of the dual problem by $\hat{\Theta}$, with the corresponding point of the original problem being given by $\hat{P} = (S + \hat{\Theta})^{-1}$. The optimal value of the dual is given by $g(\hat{\Theta})$. Because the primal problem is convex, the optimal value of the dual is the same as the optimal value of the primal: $f(\hat{P}) = g(\hat{\Theta})$, from which it can be seen

$$\rho\|\hat{P}\| + \operatorname{tr} S\hat{P} - \log|\hat{P}| = k + \log|(S + \hat{\Theta})| \tag{A.35}$$

$$\Rightarrow \rho\|\hat{P}\| = k - \operatorname{tr} S\hat{P} \tag{A.36}$$

The following result explains why the $l_1$ norm should be chosen in the construction of the original problem.

**Proposition A.3.5.** *Amongst choices of $l_p$ norm as for the penalty term, the choice $p = 1$ is the unique choice for which the resulting precision matrix has a sparse structure. Specifically, if for some $i, j$, $|\hat{\Theta}_{ij}| < \rho$, then the equivalent entry in the precision matrix has $\hat{P}_{ij} = 0$.*

*Proof.* For the optimum of a convex problem with differentiable objective function, we must have that, for all $\Theta$ in the constraint set,

$$\operatorname{tr} \nabla g(\hat{\Theta})(\Theta - \hat{\Theta}) \leq 0 \tag{A.37}$$

$$\Rightarrow \operatorname{tr} \hat{P}(\Theta - \hat{\Theta}) \leq 0 \tag{A.38}$$

which follows from differentiating $\log|S + \Theta|$ with respect to $\Theta$. Since we know that $\hat{P} \neq 0$, $\hat{\Theta}$ must lie on the edge of the constraint set, with $\|\hat{\Theta}\|_* = \rho$. The dual of the $l_1$ norm is the $l_\infty$ norm, and so the constraints of the dual problem are that $\Theta$ must lie within a box with sides at $\pm\rho$ in all dimensions.

If $|\hat{\Theta}_{ij}| < \rho$ for some $i, j$ then $\hat{\Theta}$ does not lie at a "corner" of the box. We can find an $\epsilon > 0$ for which $|\hat{\Theta}_{ij}| \leq \rho - \epsilon$, and set $\Theta^{\pm}$ to be the matrices for which all elements are identical to $\hat{\Theta}$ except for[1]

$$\Theta^{\pm}_{ij} = \hat{\Theta}_{ij} \pm \epsilon \tag{A.39}$$

Then crucially (this holds only for the $l_\infty$ norm) both $\Theta^+$ and $\Theta^-$ are in the constraint set, $\|\Theta^{\pm}\|_\infty \leq \rho$. Therefore from (A.38):

$$P_{ij}.\epsilon \leq 0 \quad \text{and} \quad P_{ij}.(-\epsilon) \leq 0 \tag{A.40}$$

$$\Rightarrow \quad P_{ij} = 0 \tag{A.41}$$

[1] And of course all the matrices are symmetric, so the same holds for $\Theta^{\pm}_{ji}$ too

□

The following results all assume that the $l_1$ norm has been chosen as for the penalty term in the original problem.

**Proposition A.3.6.** *An optimal point* $\hat{\Theta}$ *has diagonal elements all equal to* $\rho$.

*Proof.* The dual problem seeks to maximise $\log|S+\Theta|$ with $S+\Theta \succ 0$. Suppose that $\Theta$ is an optimal point, with some diagonal elements of $\Theta$ not equal to $\rho$. Then we can find a diagonal matrix $D$ with $D_{ii} \geq 0$ for every element, such that $\|\Theta + D\|_\infty \leq \rho$. Writing $U = S + \Theta$,

$$\log|U+D| = \log|U^{\frac{1}{2}}(I + U^{-\frac{1}{2}}DU^{-\frac{1}{2}})U^{\frac{1}{2}}| \tag{A.42}$$

$$= \log|U| + \sum \log(1+\lambda_i) \tag{A.43}$$

where the $\lambda_i$ are the eigenvalues of $U^{-\frac{1}{2}}DU^{-\frac{1}{2}}$. The factorisation is valid because $U \succ 0$. Since all the diagonal elements of $D$ are non-negative, $D$ is positive semidefinite, and so is $U^{-\frac{1}{2}}DU^{-\frac{1}{2}}$. So (provided $D \neq 0$) $\sum \log(1+\lambda_i) > 0$. In other words, $\Theta + D$ is a feasible point giving a higher value to the objective function than $\Theta$. This contradicts the assumption that $\Theta$ is an optimal point. □

To prove the following proposition, the following results are needed.

**Lemma A.3.7.** *If a square matrix* $A$ *is positive definite, the matrix norms satisfy*

$$\|A\|_{SV} \leq \|A\|_F \leq \|A\|_1 \tag{A.44}$$

*Proof.* Denote the eigenvalues of $A$ by $\lambda_i$. Since $A$ is positive definite, they are all positive. We have

1. $\|A\|_{SV} = \max_i \lambda_i$
2. $\|A\|_F = (\sum \lambda_i^2)^{\frac{1}{2}}$

and also

(iii) $\|A\|_1 = \sum_{ij} |A_{ij}| \geq \operatorname{tr} A = \sum \lambda_i$

The first inequality follows trivially. The second follows by observing that, for $\lambda_i \geq 0$

$$(\sum \lambda_i)^2 = \sum \lambda_i^2 + \sum_{ij} \lambda_i \lambda_j \geq \sum \lambda_i^2 \tag{A.45}$$

□

**Lemma A.3.8.** *For any $X \succ 0$,*

$$\|X\|_{SV} \leq b \Rightarrow X \preceq bI \tag{A.46}$$

*and also*

$$\|X\|_{SV} \leq b \Rightarrow X^{-1} \succeq b^{-1}I \tag{A.47}$$

*Proof.* For $X \succ 0$ the maximum singular value norm is given by the maximum eigenvalue of $X$. Therefore, for any $v \in \mathbb{R}^k$,

$$\|X\|_{SV} \leq b \Rightarrow v^T Xv \leq v^T bv = v^T bIv \tag{A.48}$$

$$\Rightarrow v^T(bI - X)v \geq 0 \tag{A.49}$$

$$\Rightarrow bI - X \succeq 0 \Rightarrow X \preceq bI \tag{A.50}$$

The condition on the singular value norm also implies that the *smallest* eigenvalue of $X^{-1}$ is greater than or equal to $b^{-1}$, so similarly

$$v^T X^{-1} v \geq v^T b^{-1} Iv \tag{A.51}$$

from which the second result follows. □

**Proposition A.3.9** (Banerjee *et al.*, 2006)**.** *For any $\rho > 0$, the optimal solution $\hat{P}$ is bounded as follows:*

$$aI \preceq \hat{P} \preceq bI \tag{A.52}$$

*where*

$$a = \frac{1}{\|S\|_{SV} + k\rho}, \qquad b = \frac{k}{\rho} \tag{A.53}$$

*Proof.* To prove the first inequality, by Lemma A.3.8, we need to show that

$$\|\hat{P}^{-1}\|_{SV} \leq a^{-1} \tag{A.54}$$

Using the triangle inequality:

$$\|\hat{P}^{-1}\|_{SV} = \|S + \hat{\Theta}\|_{SV} \leq \|S\|_{SV} + \|\hat{\Theta}\|_{SV} \tag{A.55}$$

Using Proposition A.3.7, and the fact that $\|\hat{\Theta}\|_\infty \leq \rho$,

$$\|\hat{\Theta}\| \leq \|\hat{\Theta}\|_F = (\sum_{i,j} \hat{\Theta}_{ij}^2)^{\frac{1}{2}} \leq (k^2\rho^2)^{\frac{1}{2}} = k\rho \tag{A.56}$$

and so the result is proved.

For the second result, we need that $\|\hat{P}\|_{SV} \leq b$. Using Proposition A.3.7 again, $\|\hat{P}\|_{SV} \leq \|\hat{P}\|_1$. From the condition that the dual-primal gap is zero (equation A.36) we have

$$\rho\|\hat{P}\|_1 = k - \operatorname{tr} S\hat{P} \tag{A.57}$$

But since $S \succeq 0$ and $\hat{P} \succ 0$ then $\operatorname{tr} S\hat{P} \geq 0$, and so

$$\|\hat{P}\|_1 \leq \frac{k}{\rho} \tag{A.58}$$

□

# Appendix B

# TIMIT phone recognition

## B.1 Phone recognition results

The following pages contain tables phone accuracy results on the TIMIT test data. These results are referenced in Chapter 6.

| # Gaussians per state | Phone accuracy (%) | # Params |
|---|---|---|
| 1 | 50.8 | 11,232 |
| 2 | 54.1 | 22,464 |
| 4 | 56.9 | 44,928 |
| 6 | 58.9 | 67,392 |
| 8 | 59.5 | 89,856 |
| 10 | 61.0 | 112,320 |
| 12 | 61.8 | 134,784 |
| 16 | 63.0 | 179,712 |
| 20 | 63.2 | 224,640 |
| 24 | 63.7 | 269,568 |
| 28 | 63.8 | 314,496 |
| 32 | 64.5 | 359,424 |
| 36 | 64.4 | 404,352 |
| 40 | 64.6 | 449,280 |
| 44 | 64.4 | 494,208 |
| 48 | 64.5 | 539,136 |
| 52 | 65.0 | 584,064 |
| 56 | 65.2 | 628,992 |
| 60 | 65.2 | 673,920 |
| 64 | 65.6 | 718,848 |
| 68 | 65.8 | 763,776 |
| 72 | 66.0 | 808,704 |
| 76 | 65.7 | 853,632 |
| 80 | 66.0 | 898,560 |
| 84 | 65.7 | 943,488 |
| 88 | 65.7 | 988,416 |
| 92 | 65.7 | 1,033,344 |
| 96 | 65.4 | 1,078,272 |
| 100 | 65.3 | 1,123,200 |

Table B.1: Phone accuracy of diagonal covariance models trained on the full training set, also showing the number of mean and variance parameters.

| | Proportion of full training set used | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| # Gaussians | 10% | 20% | 30% | 40% | 50% | 60% | 75% | 100% |
| 1 | 50.3 | 50.5 | 51.0 | 50.6 | 50.6 | 50.8 | 50.5 | 50.8 |
| 2 | 53.5 | 53.2 | 53.9 | 54.4 | 53.8 | 53.6 | 54.2 | 54.1 |
| 4 | 55.8 | 57.7 | 57.4 | 57.6 | 57.1 | 57.6 | 57.5 | 56.9 |
| 8 | 57.6 | 59.8 | 59.5 | 59.3 | 59.3 | 59.3 | 59.8 | 59.5 |
| 12 | 58.7 | 60.1 | 61.7 | 61.3 | 61.0 | 61.6 | 61.9 | 61.8 |
| 16 | 58.2 | 60.2 | 61.9 | 61.5 | 62.2 | 62.0 | 63.0 | 63.0 |
| 20 | 58.5 | 60.9 | 61.7 | 62.6 | 62.5 | 62.4 | 63.4 | 63.2 |
| 24 | 57.8 | 61.0 | 61.5 | 63.1 | 62.7 | 63.3 | 64.1 | 63.7 |
| 32 | - | 61.0 | 61.9 | 63.9 | 63.4 | 63.4 | 64.5 | 64.5 |
| 40 | - | 60.9 | 61.7 | 63.9 | 63.6 | 63.8 | 64.9 | 64.6 |
| 48 | - | - | 61.4 | 63.7 | 64.3 | 64.4 | 65.1 | 64.5 |
| 56 | - | - | 61.5 | 62.8 | 63.8 | 64.4 | 64.9 | 65.2 |
| 64 | - | - | - | 63.1 | 63.6 | 64.5 | 65.2 | 65.6 |
| 72 | - | - | - | 63.0 | 63.1 | 64.0 | 64.8 | 66.0 |
| 80 | - | - | - | - | 63.4 | 63.5 | 64.6 | 66.0 |

Table B.2: Phone accuracy of diagonal covariance models trained on subsets of the full training set.

| | Full data | | 50% data | | 10% data | |
|---|---|---|---|---|---|---|
| $\rho$ | Acc | # params | Acc | # params | Acc | # params |
| 0.0 | 67.2 | 1,415,232 | 65.1 | 1,415,232 | 41.1 | 1,415,232 |
| 0.005 | 66.2 | 1,401,049 | 64.7 | 1,398,878 | 45.9 | 1,409,197 |
| 0.01 | 65.9 | 1,381,307 | 64.5 | 1,383,103 | 47.4 | 1,401,745 |
| 0.02 | 65.6 | 1,343,641 | 64.3 | 1,347,198 | 49.1 | 1,379,110 |
| 0.03 | 65.4 | 1,320,814 | 64.1 | 1,322,443 | 50.4 | 1,357,421 |
| 0.04 | 65.1 | 1,307,283 | 63.9 | 1,308,283 | 51.5 | 1,341,786 |
| 0.05 | 64.8 | 1,298,833 | 63.8 | 1,298,287 | 52.0 | 1,328,355 |
| 0.06 | 64.5 | 1,292,878 | 63.7 | 1,292,208 | 52.3 | 1,319,076 |
| 0.1 | 63.7 | 1,275,523 | 63.4 | 1,274,459 | 54.2 | 1,293,674 |
| 0.15 | 63.2 | 1,253,411 | 62.7 | 1,253,756 | 54.8 | 1,276,112 |
| 0.2 | 62.8 | 1,231,128 | 62.3 | 1,234,122 | 55.2 | 1,261,418 |
| 0.3 | - | - | - | - | 55.8 | 1,235,100 |
| 0.4 | - | - | - | - | 55.8 | 1,212,036 |
| 0.5 | - | - | - | - | 56.1 | 1,190,211 |
| 0.6 | - | - | - | - | 55.7 | 1,165,171 |
| 0.8 | - | - | - | - | 55.7 | 1,107,059 |
| 1.0 | - | - | - | - | 55.2 | 1,039,944 |
| 1.5 | - | - | - | - | 54.3 | 871,635 |
| 2.0 | - | - | - | - | 54.0 | 722,542 |
| Semi tied | 64.5 | 353,808 | 63.7 | 353,808 | 53.0 | 353,808 |
| Diagonal | 61.8 | 134,784 | 61.0 | 134,784 | 58.7 | 134,784 |

Table B.3: Phone accuracy of sparse GM models with 12 Gaussians per state, varying penalty parameter, showing the number of Gaussian parameters. Mean parameters are included, although they are not reduced to zero by the penalisation.

| | Proportion of full training set used | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Prior $\tau$ | 10% | 20% | 30% | 40% | 50% | 60% | 75% | 100% |
| 0 | 41.1 | 56.4 | 61.8 | 63.5 | 65.1 | 65.6 | 66.6 | 67.2 |
| 25 | 59.1 | 62.9 | 64.3 | 64.9 | 66.5 | 66.5 | 67.1 | 67.5 |
| 50 | 59.6 | 63.3 | 64.4 | 64.9 | 66.3 | 66.5 | 67.2 | 67.2 |
| 100 | 59.5 | 62.7 | 64.2 | 64.7 | 65.8 | 66.3 | 66.9 | 67.0 |
| 150 | 59.5 | 62.6 | 63.9 | 64.5 | 65.6 | 66.3 | 66.3 | 66.9 |

Table B.4: Phone accuracy of covariance models with 12 Gaussians per state, with varying amounts of training data

| # Gaussians | Diagonal | Semi-tied | Naive full | Shrinkage |
|---|---|---|---|---|
| 1 | 50.8 | 57.1 | 58.0 | 58.2 |
| 2 | 54.1 | 60.6 | 62.2 | 62.0 |
| 4 | 56.9 | 62.6 | 63.9 | 64.1 |
| 6 | 58.9 | 63.3 | 65.8 | 65.7 |
| 8 | 59.5 | 63.6 | 65.7 | 65.9 |
| 10 | 61.0 | 64.6 | 66.4 | 66.6 |
| 12 | 61.8 | 64.5 | 67.2 | 67.2 |
| 16 | 63.0 | 64.9 | 67.3 | 67.2 |
| 20 | 63.2 | 64.8 | 66.6 | 67.0 |
| 24 | 63.7 | 65.7 | 66.9 | 67.5 |
| 28 | 63.8 | 65.9 | 66.6 | 67.8 |
| 32 | 64.5 | 65.5 | 65.5 | 67.1 |
| 36 | 64.4 | 66.5 | 65.0 | 67.1 |
| 40 | 64.6 | 66.3 | 64.2 | 67.1 |
| 44 | 64.4 | 66.5 | 63.8 | 67.3 |
| 48 | 64.5 | 66.4 | 62.9 | 67.1 |

Table B.5: Phone accuracy (%) of covariance models with varying number of Gaussians, trained on the full training set.

| # Gaussians | Diagonal | Semi-tied | Naive full | Shrinkage |
|---|---|---|---|---|
| 1 | 50.6 | 56.6 | 57.4 | 57.4 |
| 2 | 53.8 | 60.6 | 61.2 | 61.4 |
| 4 | 57.1 | 61.9 | 62.9 | 63.1 |
| 6 | 59.3 | 62.9 | 64.7 | 64.6 |
| 8 | 59.3 | 63.4 | 64.6 | 64.8 |
| 10 | 60.5 | 63.5 | 65.1 | 65.6 |
| 12 | 61.0 | 63.7 | 65.1 | 66.4 |
| 16 | 62.2 | 64.2 | 64.7 | 66.2 |
| 20 | 62.5 | 64.6 | 63.5 | 65.9 |
| 24 | 62.7 | 64.4 | 61.8 | 65.9 |
| 28 | 63.1 | 64.9 | 60.1 | 66.0 |
| 32 | 63.4 | 64.6 | 57.3 | 65.2 |
| 36 | 63.5 | 64.5 | 55.6 | 65.5 |
| 40 | 63.6 | 64.6 | 54.1 | 65.3 |
| 44 | 63.5 | 64.6 | 52.6 | 64.4 |
| 48 | 64.3 | 64.8 | 51.1 | 64.4 |

Table B.6: Phone accuracy (%) of covariance models with varying number of Gaussians, trained on the 50% training set.

| # Gaussians | Diagonal | Semi-tied | Naive full | Shrinkage |
|---|---|---|---|---|
| 1 | 50.3 | 54.6 | 56.1 | 56.4 |
| 2 | 53.5 | 56.0 | 58.4 | 59.2 |
| 4 | 55.8 | 55.3 | 56.3 | 59.8 |
| 6 | 56.7 | 55.0 | 53.7 | 60.3 |
| 8 | 57.6 | 54.7 | 49.2 | 60.1 |
| 10 | 57.8 | 53.3 | 45.0 | 59.8 |
| 12 | 58.7 | 53.0 | 41.1 | 59.2 |

Table B.7: Phone accuracy (%) of covariance models with varying number of Gaussians, trained on the 10% training set.

| # Gaussians | Diagonal | Semi-tied | Full covariance |
|---|---|---|---|
| 1 | 11,232 | 230,256 | 117,936 |
| 2 | 22,464 | 241,488 | 235,872 |
| 4 | 44,928 | 263,952 | 471,744 |
| 6 | 67,392 | 286,416 | 707,616 |
| 8 | 89,856 | 308,880 | 943,488 |
| 10 | 112,320 | 331,344 | 1,179,360 |
| 12 | 134,784 | 353,808 | 1,415,232 |
| 16 | 179,712 | 398,736 | 1,886,976 |
| 20 | 224,640 | 443,664 | 2,358,720 |
| 24 | 269,568 | 488,592 | 2,830,464 |
| 28 | 314,496 | 533,520 | 3,302,208 |
| 32 | 359,424 | 578,448 | 3,773,952 |
| 36 | 404,352 | 623,376 | 4,245,696 |
| 40 | 449,280 | 668,304 | 4,717,440 |
| 44 | 494,208 | 713,232 | 5,189,184 |
| 48 | 539,136 | 758,160 | 5,660,928 |

Table B.8: Number of mean and variance parameters for three types of covariance model, with varying numbers of Gaussians.

## B.2 Phone mappings

| 39-phone set | 48-phone set |
|---|---|
| b | b |
| d | d |
| g | g |
| p | p |
| t | t |
| k | k |
| dx | dx |
| jh | jh |
| ch | ch |
| s | s |
| z | z |
| zh | zh, sh |
| f | f |
| th | th |
| v | v |
| dh | dh |
| m | m |
| n | n, en |
| ng | ng |
| r | r |
| w | w |
| y | y |
| hh | hh |
| el | el, l |
| iy | iy |
| eh | eh |
| ey | ey |
| ae | ae |
| aa | aa, ao |
| aw | aw |
| ay | ay |
| oy | oy |
| ow | ow |
| uh | uh |
| uw | uw |
| er | er |
| ax | ax, ah |
| ix | ix, ih |
| sil | sil, cl, vcl, epi |

Table B.9: Comparing the 48-phone set used for acoustic modelling with the 39-phone set used to obtain phone accuracy scores.