# CONDITIONAL RANDOM FIELDS FOR CONTINUOUS SPEECH RECOGNITION

By

Yasser Hifny Abdel-Haleem

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
FACULTY OF ENGINEERING
UNIVERSITY OF SHEFFIELD
SHEFFIELD, UK
NOVEMBER 2006

# Abstract

Acoustic modelling based on Hidden Markov Models (HMMs) is employed by state-of-the-art stochastic speech recognition systems. Although HMMs are a natural choice to warp the time axis and model the temporal phenomena in the speech signal, they do not model the spectral phenomena well. This is a consequence of their conditionally independent properties, which are inadequate for sequential processing.

In this work, a new acoustic modelling paradigm based on Conditional Random Fields (CRFs) is investigated and developed. This paradigm addresses some of the weak aspects of HMMs while maintaining many of the good aspects, which have made them successful. In particular, the acoustic modelling problem is reformulated in a data driven, sparse, augmented space to increase discrimination. In addition, acoustic context modelling is explicitly integrated to handle the sequential phenomena of the speech signal. In a phone recognition task, test results have shown significant improvements over comparable HMM systems.

While the theoretical motivations behind this work are attractive, a practical implementation of these complex systems is demanding. The main goal of this work is to present an efficient framework for estimating these models that ensures scalability and generality for large vocabulary speech recognition systems.

# Acknowledgements

أَنِ اشْكُرْ لِى وَلِوَالِدَيْكَ

*Thank Me and your parents*

This thesis could not have been completed without advice and support from my advisor Steve Renals. I am grateful to him for allowing me the freedom to explore many new ideas with endless patience and constructive criticism. His sincerity, integrity, and intelligence are much appreciated.

During my research, I have had the pleasure of working in the multidisciplinary environment provided by the SPandH group. I would like to thank Prof. Phil Green and all the past and current members of the group for their support. Thanks to Yoshi Gotoh, who followed my work during Steve's sabbatical visit to ICSI, and Neil Lawrence, who was an official advisor after Steve's move to Edinburgh. I am especially grateful to Andre Coy for proof reading large parts of this thesis. Thanks to Tonatiuh Pena for his comments on Chapter 4.

I would like to acknowledge that my graduate studies were supported in part by a Motorola scholarship award. I would like to thank David Peace for his support and help.

My love and thanks go to my parents for their unwavering encouragement and support during this work. May *Allah* bless and reward all those who helped me.

# Table of Contents

# Chapter 1

# Introduction

Despite ongoing improvement in recognition accuracy, automatic speech recognition is still a challenging task. The use of statistical methods has been the major contributing factor to the current success of developing automatic speech recognition systems. In such methods, large corpora of annotated speech are used to train acoustic models to capture the underlying variability of speech. These models are used to decode a speech signal. However, it is generally agreed that the underlying technologies behind the acoustic models may be the main limitation to the achievement of human-like recognition performance. With the availability of larger databases and computational power, a possibility to scale acoustic models to large numbers of parameters and to estimate them from data has proven to be very powerful. Scaling acoustic models may provide a means of integrating additional prior information related to speech production, which conceptually may improve speech recognition.

State-of-the-art acoustic models are designed to handle a sequence of acoustic vectors in a $d$-dimensional space, which are robust for speech variability. In order to model the time variations of the speech signal, Hidden Markov Models (HMMs) were a natural choice to warp the time axis and model the temporal phenomena in the speech signal. Spectral variability is usually modelled by mixtures of Gaussian densities. Assuming the shape (i.e. parametric) of the stochastic process generation, usually leads to well understood models and efficient training methods.

Some recent advances in acoustic modelling research are based on discriminative feature projection [Povey, 2005, Povey et al., 2005, Morgan et al., 2005]. The new feature sets may be designed to overcome the basic HMM limitations or to improve the discrimination between speech classes given some prior knowledge about the speech production system. This is usually achieved without changing the basic HMM acoustic modelling framework. Within the HMM framework, improved discriminative training techniques were developed based on the overall risk criterion estimation [Kaiser et al., 2002], large margin estimation and its variants [Li et al., 2005, Liu et al., 2005], and soft margin estimation [Li et al., 2006]. In addition, discriminative training techniques have been scaled to large vocabulary speech recognition [Woodland and Povey, 2000, Povey, 2004]. Beyond the HMM framework, many models have been proposed with limited success to improve the acoustic modelling component.

In this thesis, we took another approach to improve the acoustic modelling component. Rather than removing the HMM model and introduce another model and check its recognition performance, we design an acoustic model closely related to the HMM framework. To improve the discrimination between speech classes, the new framework will integrate basic pattern classification and sequential processing concepts. Hence, the main goal is to design an acoustic model to *discriminate* between speech classes rather than to propose an acoustic model, which aims to *generate* the acoustic observation as widespread within the acoustic modelling research.[1]

To focus on a predefined goal (improved discrimination), the acoustic modelling problem is reformulated in a high dimensional space to increase the discrimination between confusable classes. This step is followed by modelling and integrating the acoustic context information into the high dimensional space to model the sequential phenomena of the speech signal. Once these augmented spaces are constructed, we select a linear chain Conditional Random Field (CRF) model as our basic acoustic model as it is a flexible tool for handling this new formulation and its computational complexity is very similar to an HMM. Linear chain CRFs are undirected graphical models formulated using the Maximum Entropy (MaxEnt) principle. These models can be thought as the nonparametric twins for HMMs (see Chapter 5). The overall formulation will lead to sparse context modelling in a high dimensional space ($\approx 10^6$ dimensions), which is a fundamental concept inherited from pattern classification and sequential processing used to improve speech recognition in general. Our treatment can be considered as a novel extension to the current HMM systems as the new system focuses on integrating basic concepts that conceptually can improve speech recognition system.

This new paradigm addresses some of the weak aspects of HMMs while maintaining many of the good aspects, which have made them successful. In particular, the proposed system has similar training speed and similar decoding algorithms, which may be the key ideas behind the success of the HMMs. In addition, the new system can be initialized from any state-of-the-art speech recognition system based on the

[1] The success of HMMs as generative models has motivated the design and development of complex generative models. However, generative modelling is not directly related to the discrimination between speech classes. Consequently, we argue that having good generative speech models are useful for speech synthesis but not necessarily for speech recognition.

HMM technology. On the other hand, moving to high dimensional spaces may limit the scalability of the new approach for large vocabulary speech recognition systems. Hence, a key idea behind the new system is a scalable discriminant compression algorithm, which allows the system to integrate the acoustic context in the augmented spaces and offers full control to prune the parameter space without any computational cost during the training process.

In the following sections, some basic elements related to the development of this thesis are introduced. In particular, a general review of speech recognition, information theory, and graphical modelling is presented.

## 1.1 Automatic Speech Recognition

Automatic speech recognition systems, also known as speech-to-text systems, are the core technology for man-machine interface. These systems aim to find the most likely word sequence given acoustic observations collected from a speech signal. Using statistical methods [Jelinek, 1997], speech recognition can be defined as a problem of choosing a word sequence $\hat{\mathbf{W}}$ with the *maximum a posterior* (MAP) criterion given a time sequence of speech frames or acoustic observations associated an utterance $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{O}) \tag{1.1.1}$$

Using Bayes's rule, equation (1.1.1) can be written as

$$\begin{aligned} P(\mathbf{W}|\mathbf{O}) &= \frac{p(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{p(\mathbf{O})} \\ &\propto p(\mathbf{O}|\mathbf{W})P(\mathbf{W}) \end{aligned} \tag{1.1.2}$$

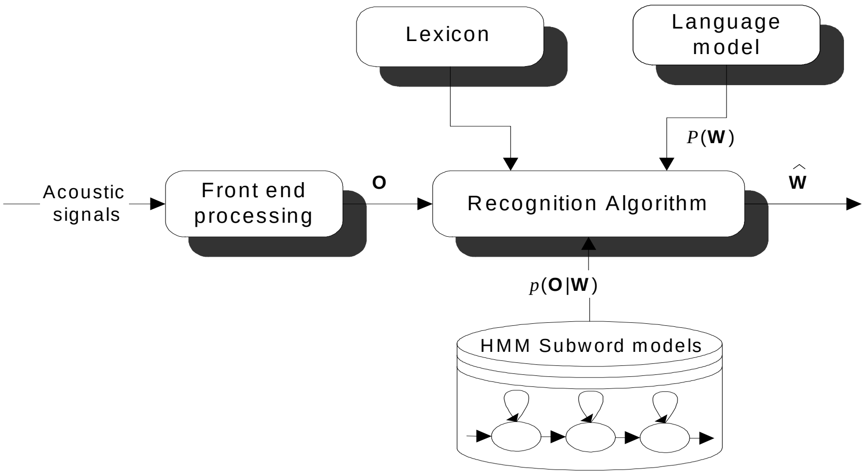


Figure 1.1: General speech recognition system.

where $p(\mathbf{O}|\mathbf{W})$ is the likelihood of the acoustic observations given by an *acoustic model* and $P(\mathbf{W})$ is the likelihood of the hypothesized word sequence given by a *language model*.

State-of-the-art acoustic modelling is based on HMM models to warp the time axis since the speech signal is not a static pattern. HMM models will be detailed in Chapter 2. Acoustic models usually are based on subword HMMs, which are trained using a corpus of transcribed speech data. To generate a sentence hypothesis, the word and sentence level HMMs are constructed via a *lexicon*, which describes the possible pronunciation of each word allowable in the recognition task. Large vocabulary speech recognition systems (LVCSR) use a *n-gram* language model, which gives an approximate probability score of an allowable word sequence $\mathbf{W} = (w^{(0)}w^{(1)} \dots w^{(m)})$ in the recognition task. This probability score is calculated by accumulating local scores using $n-1$ Markov chains over the word sequence and is given by

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w^{(i)}|w^{(i-n+1)} \dots w^{(i-1)}) \tag{1.1.3}$$

The relationship between speech recognition components is shown in figure 1.1. During the recognition process, efficient search algorithms are employed to find the most probable word sequence $\hat{\mathbf{W}}$ given the acoustic observations extracted via front end processing. The performance of a speech recognition system is usually measured with the Word Error Rate (WER) criterion, which is based on a dynamic string alignment algorithm between a recognized word sequence against the correct (reference) word sequence [Levenshtein, 1965]. The WER criterion is derived from the Levenshtein (string edit) distance and is defined as

$$\text{WER} = \frac{\text{Sub} + \text{Del} + \text{Ins}}{\#\text{words in the reference sentence}} \times 100\% \tag{1.1.4}$$

where the number of substitutions (Sub), deletions (Del), and insertions (Ins) is computed with an alignment between the reference and hypothesis strings.

Measuring the performance of a speech recognition system based on WER criterion as in equation (1.1.4) is not consistent with the MAP decoding criterion in equation (1.1.1), which minimizes the *sentence error rate*. This may lead to suboptimal decoding results. Hence, a decoding criterion that aims to measure the expected WER directly via confusion networks is used in recent decoding algorithms [Mangu et al., 1999, Mangu et al., 2000] and it is given by

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{\mathbf{R}} P(\mathbf{R}|\mathbf{O})\text{WE}(\mathbf{W}, \mathbf{R}) \tag{1.1.5}$$

where $\text{WE}(\mathbf{W}, \mathbf{R})$ is string edit distance between a hypothesis string $\mathbf{W}$ and a reference string $\mathbf{R}$.

In recent years, dramatic advances have been made in automatic speech recognition and many tasks have been evaluated. For some tasks[2] [Lippmann, 1997], Table

[2] Note that, the machine error for Switchboard task shows the current state-of-the-art systems, which yield WER within a range of 20-30%.

1.1 summaries speech recognition performance by machines and human measured in WER. The performance of speech recognition systems clearly degrades in the adverse acoustic conditions associated with a spontaneous telephone conversations (Switchboard) task. Towards integrating speech recognition in multimodal applications, transcription of speech in meeting and lectures may be the main goal of the current evaluation tasks.

Table 1.1: *Comparison between human and machine recognition performance.*

| Corpus | Description | Vocabulary size | Machine error | Human error |
|---|---|---|---|---|
| TI digits | Read digits | 10 | 0.72% | 0.009% |
| Alphabet letters | Read alphabet | 26 | 5% | 1.6% |
| Resource Management | Read sentences | 1000 | 3.6% | 0.1% |
| Business News | Read sentences | 5000-Unlimited | 7.2% | 0.9% |
| Switchboard | Spontaneous telephone | 2000-Unlimited | 20-30% | 4% |

## 1.2 Information Theory for Pattern Recognition

This thesis is developed in the context of the Maximum Entropy principle. Hence, basic elements of information theory are introduced (for details see [Gallager, 1968, Cover and Thomas, 1991, MacKay, 2003]). In his landmark paper [Shannon, 1948], Shannon introduced a quantitative measure of information known as *entropy.* For a discrete random variable $\mathbf{s}$ takes values $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ with the corresponding probabilities $P(\mathbf{s}_1), P(\mathbf{s}_2), \ldots, P(\mathbf{s}_n)$, the entropy is defined as

$$S(\mathbf{s}) \triangleq -\sum_{i=1}^{n} P(\mathbf{s}_i) \log P(\mathbf{s}_i) \tag{1.2.1}$$

The entropy is a measure of the average uncertainty of a random variable and $-\log P(\mathbf{s}_i)$ is the amount of information gained by observing the event $\mathbf{s}_i$. Events $\mathbf{s}_i$

with low probabilities produce more information than the events with large probabilities. Hence, rare events produce more information or surprise than frequent events and this is the basic idea behind compression algorithms. The entropy has large values when all $\mathbf{s}_i$ have same probability. The entropy is measured in *nats* when the natural logarithm is used in equation (1.2.1) to measure the information conveyed by a random variable or in *bits* when base 2 logarithm is used.

Similarly, the *conditional entropy* is the uncertainty in a random variable $\mathbf{s}$ given another random variable $\mathbf{o}$ and it is given by

$$S(\mathbf{s}|\mathbf{o}) \triangleq -\sum_{\mathbf{s},\mathbf{o}} P(\mathbf{s},\mathbf{o}) \log P(\mathbf{s}|\mathbf{o}) \tag{1.2.2}$$

The average *mutual information* is defined as the difference between $S(\mathbf{s})$ and $S(\mathbf{s}|\mathbf{o})$ as shown in equation (1.2.3). It is a measure of average reduction in uncertainly about $\mathbf{s}$ after observing $\mathbf{o}$. The mutual information is symmetric $I(\mathbf{s};\mathbf{o}) = I(\mathbf{o};\mathbf{s})$ and is always nonnegative.

$$\begin{aligned} I(\mathbf{s};\mathbf{o}) &\triangleq S(\mathbf{s}) - S(\mathbf{s}|\mathbf{o}) \\ &= -\sum_{\mathbf{s},\mathbf{o}} P(\mathbf{s},\mathbf{o}) \log \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{s})P(\mathbf{o})} \end{aligned} \tag{1.2.3}$$

Relative entropy or the *Kullback-Leibler* (KL) divergence between two probability distributions $P$ and $Q$ of a discrete random variable is given by

$$\mathrm{D_{KL}}(P||Q) \triangleq \sum_{\mathbf{s}} P(\mathbf{s}) \log \frac{P(\mathbf{s})}{Q(\mathbf{s})} \tag{1.2.4}$$

$\mathrm{D_{KL}}(P||Q)$ is the information loss when $Q$ is used to approximate $P$. KL divergence is nonsymmetric (i.e. $KL(P||Q) \neq KL(Q||P)$) and $\mathrm{D_{KL}}(P||Q) \geq 0$. When the distributions $P$ and $Q$ are identical, $\mathrm{D_{KL}}(P||Q)$ is exactly zero. The mutual information is the KL divergence between the joint distribution $P(\mathbf{s},\mathbf{o})$ and the product of $P(\mathbf{o})$ and $P(\mathbf{s})$ distributions.

In statistical pattern recognition, $Q = \tilde{P}_\Lambda$ is an hypothesized model that has free parameters $\Lambda$. The goal of the training process is to minimize the information loss in terms of KL divergence between the training data distribution[3] $\tilde{P}$ and its hypothesized model $\tilde{P}_\Lambda$ (i.e. $\mathrm{D}_{\mathrm{KL}}(\tilde{P}||\tilde{P}_\Lambda)$ is minimum). Hence, $\Lambda^*$ is given by

$$\begin{aligned}
\Lambda^* &= \arg\min_\Lambda \{\mathrm{D}_{\mathrm{KL}}(\tilde{P}||\tilde{P}_\Lambda)\} \\
&= \arg\min_\Lambda \{\sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \tilde{P}(\mathbf{s}) - \sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \tilde{P}_\Lambda(\mathbf{s})\} \\
&= \arg\min_\Lambda \{S(\tilde{P}) - \sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \tilde{P}_\Lambda(\mathbf{s})\} \\
&\propto \arg\min_\Lambda \{-\sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \tilde{P}_\Lambda(\mathbf{s})\} = \arg\min_\Lambda \{S(\tilde{P}, \tilde{P}_\Lambda)\}
\end{aligned} \tag{1.2.5}$$

where $S(\tilde{P}, \tilde{P}_\Lambda)$ is defined as the *cross entropy* between two distribution $\tilde{P}$ and $\tilde{P}_\Lambda$ and the term $S(\tilde{P})$ is ignored because it is independent of $\Lambda$. As a result, the minimization of $S(\tilde{P}, \tilde{P}_\Lambda)$ and $\mathrm{D}_{\mathrm{KL}}(\tilde{P}||\tilde{P}_\Lambda)$ are equivalent. The minimization of the cross entropy between a data model and an hypothesized model is equivalent to the maximization of the log likelihood objective function, which is given by

$$\begin{aligned}
\Lambda^* &= \arg\max_\Lambda \mathcal{L}(\tilde{P}, \tilde{P}_\Lambda) = \arg\max_\Lambda \sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \tilde{P}_\Lambda(\mathbf{s}) \\
&= \arg\max_\Lambda -S(\tilde{P}, \tilde{P}_\Lambda)
\end{aligned} \tag{1.2.6}$$

Hence, minimization of KL objective function between a data model and an hypothesized model is related the maximum likelihood estimation (MLE) between the two models. Similarly, maximizing the mutual information can be re-cast as minimizing the cross entropy between a data model and an hypothesized model [Rabiner, 1989].

[3] A true distribution $P$ that generates a data set is usually not known and is replaced with an empirical distribution $\tilde{P}$ observed from the stochastic process.

## 1.3 Graphical Modelling

A graphical model is a probabilistic graph $G = (V, E)$ where the nodes or vertices $V = v_1, \ldots, v_n$ represent random variables and the edges $E$ represent a group of conditional independence properties or relations between the random variables [Lauritzen, 1996]. One of the major advantages of graphical models is that statistical models such as Factor Analysis (FA), principal component analysis (PCA), mixtures of Gaussian models (GMMs) , vector quantization (VQ), Kalman filter models (linear dynamical system), and HMMs can be interpreted as variations of a single graphical model [Roweis and Ghahramani, 1999].

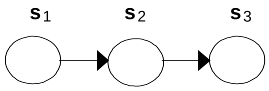


Figure 1.2: An example of a directed graph describing the joint probability $P(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$.

Directed graphical models or Bayesian Networks [Pearl, 1988] factorize the joint probability distribution over the random variables $V$ as:

$$P(V) = \prod_i P(v_i|\text{pa}(v_i)) \tag{1.3.1}$$

where $\text{pa}(v_i)$ are the parents of $v_i$ in the graph $G$. The random variables $V$ are connected by a directed acyclic graph, where the edges $E$ specify conditional independence relations between the variables. Sequential or temporal processes are modelled by a variant referred to as Dynamic Bayesian Networks (DBNs). Figure 1.2 shows an example of a joint probability, $P(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$, over a chain defined by a graph, which arises in an HMM. This joint probability can be factorized:

$$P(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3) = P(\mathbf{s}_1)P(\mathbf{s}_2|\mathbf{s}_1)P(\mathbf{s}_3|\mathbf{s}_2) \tag{1.3.2}$$

where the links define the conditional independence properties and the arrows define the nodes's parent-child relationship (e.g. the node associated with $\mathbf{s}_2$ is the parent of the node associated with $\mathbf{s}_3$). HMMs may be regarded as a special, constrained case of DBNs, but recent research started to look at less-constrained, more flexible DBNs for speech recognition [Zweig and Russell, 1998, Bilmes and Zweig, 2002, Bilmes and Bartels, 2005]. DBNs can offer a flexible statistical framework for developing speech recognition systems, where a frame can be associated with many random variables representing a complex stochastic production of a speech signal.

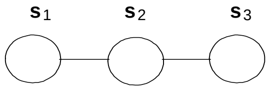


Figure 1.3: An example of a linear chain undirected graph.

Markov Random Fields (MRFs) [Kinderman and Snell, 1980] or undirected graphical models factorize the joint distribution as a simple product of clique functions given by

$$P(V) = \frac{1}{Z} \prod_{\text{cliques } c} \phi_c(v_c) \tag{1.3.3}$$

where $c$ is a set of cliques of $G$ and $v_c$ is a set of variables in clique $c$. A clique is a set of vertices (nodes), which defines a neighborhood system such that there exists an edge connecting every two vertices in $c$ (i.e. the nodes are fully connected). Moreover, a maximal clique is a clique that has the largest number of variables over $G$ and a graph can have more than one maximal clique. $Z = \sum_V \prod_{\text{cliques } c} \phi_c(v_c)$, commonly called the partition function, is a normalization constant. For example, the graph in Figure 1.3 has two cliques of two nodes given by $\{\mathbf{s}_1, \mathbf{s}_2\}$, $\{\mathbf{s}_2, \mathbf{s}_3\}$ and the edges are undirected (no arrows). The joint probability can be written as

$$P(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3) = \frac{1}{Z}\phi_1(\mathbf{s}_1, \mathbf{s}_2)\phi_2(\mathbf{s}_2, \mathbf{s}_3) \tag{1.3.4}$$

The value of a clique function or potential $\phi_c$ is nonnegative and a common form of potentials, which leads to the e-family distributions, is given by

$$\phi_c(v_c) = \exp(\sum_i \lambda_i f_i(v_c)) \tag{1.3.5}$$

where $\lambda_i$ are the potential parameters and $f_i(v_c)$ are sufficient statistics (constraints) observed from a stochastic process. This thesis formulates the acoustic modelling problem in the context of sequential MRF modelling; thus equation (1.3.3) will be revisited in a conditional form in the following chapters.

Directed graphical models are used for causal modelling in statistics and artificial intelligence while undirected graph models or MRFs are more frequently used in image processing and statistical physics to model correlational or spatial information [Jordan and Sejnowski, 2001]. In general, it is possible to convert directed graphical models into undirected graphical models. This may be useful to initialize undirected graphical models from estimated directed graphical models (see Chapter 5).

## 1.4 Thesis Overview

This work aims to introduce an acoustic modelling component of a continuous speech recognition system.

The acoustic modelling problem is extensively detailed in the literature. Hence, Chapter 2 primarily focuses on the relevant research related to the work described in this thesis. In Chapter 3, a mathematical treatment for the principle of Maximum

Entropy is presented. A new family of approximate iterative scaling algorithms to estimate the MaxEnt/CRF models will be derived. Chapter 4 will develop an acoustic classifier formulation based on kernel spaces, which will lead to a kernel machine based on CRFs. Chapter 5 presents an identical nonparametric formulation of the HMM graphical model as a CRF model. A new parameter estimation method based on approximate iterative scaling will be also presented. The suggested model will allow direct evaluation of sequential CRFs for the TIMIT phone recognition task given similar HMMs. A new formulation for the acoustic modelling problem is presented in Chapter 6. The new design inherits some fundamental concepts from pattern classification and sequential processing with the aid of a flexible CRF modelling to improve speech recognition. Finally, the main ideas and conclusions of the thesis are summarized with some suggestions for further work.

# Chapter 2

# The Acoustic Modelling Problem

Acoustic modelling aims to compute the probabilities of the acoustic observations given a possible word sequence during the search operation. The quality of an acoustic model is measured by its ability to discriminate between phonetic units, which generate a sentence hypothesis via a lexicon. The discrimination between phonetic units is a *challenging* task given the variability in the speech signal, context, speaker, and environment.

In this chapter, we review some major elements that contribute to the quality of the acoustic modelling component. Other components, which contribute directly to the quality of a recognition system such as the language model and search algorithms are detailed in [Huang et al., 2001, Jelinek, 1997]. There is a mutual relationship between feature extraction from the speech signal and the acoustic modelling design and quality. Hence, feature extraction for speech recognition will be introduced in the next section. HMM based acoustic modelling is the choice for state-of-the-art stochastic speech recognition engines. On the other hand, the acoustic models developed in this thesis are based on CRFs, which are conceptually similar to the HMM models. In fact, a linear chain CRF model is actually the nonparametric twin of an

HMM and both models have similar inference. However, it is important to mention that our work focuses on an implementation of some concepts that aimed at increasing discrimination between speech classes inherited from pattern classification and sequential processing rather than replacing the HMM with its nonparametric twin CRF model. The use of the CRF model is motivated by its flexible implementation of the suggested ideas.

HMM model definition, inference, and parameter estimation will be reviewed in Section 2.2. Several models have been proposed to improve acoustic modelling beyond the basic HMM model. These attempts will be highlighted in Section 2.3. However, HMM based acoustic modelling is still the main stream speech recognition technology and no other model significantly and consistently outperforms the HMM.

In Section 2.4, a general approach, which can improve acoustic modelling and speech recognition in general will be reviewed. This approach, which is called *speech recognition committee*, solves complex tasks by handling large numbers of simple tasks in a similar way to the divide-and-conquer algorithms.

## 2.1 Acoustic Representation

Speech signal processing assumes the speech signal is a *piecewise* stationary signal. As a result, a pre-processor converts the speech signal into a sequence of speech frames or acoustic observations using short time signal analysis. Typically, these frames are calculated every 10-20ms from 20-30ms windows of speech. Speech frames of these lengths are short enough that the estimated parameters can be assumed constant within each frame. Feature extraction for speech recognition problems aims to find the intrinsic information related to vocal tract shape, which may be considered invariant among all speakers (i.e. invariant acoustic space). A feature vector extracted from a frame contains a set of *independent* features representing the *envelope* of the speech spectrum. The basic assumption behind this idea is that the envelope of the spectrum is a *course* representation of the spectrum that has all relevant information related to the speech recognition problem. In general, the *fine* spectral structure contains information about the excitation (i.e. details related to speakers or voicing) in a source-filter speech production model, which may be a source of noise for speech recognizers [Rabiner and Juang, 1993, Gold and Morgan, 1999].

The perceptually motivated front end processing based on Mel frequency Cepstral Coefficients (MFCCs) is the most widely used feature extraction method for speech recognition systems [Davis and Mermelstein, 1980]. MFCCs are derived from the short time Fourier transform of the speech signal. The spectrum is warped according to the Mel frequency scale, which is based on studies of human auditory perception. Finally, the cepstrum is computed by taking the inverse discrete Cosine Transform (DCT) to construct a feature vector representing a speech frame. The extracted features using DCT projection approximates a Karhunen-Loéve transformation for a first order Gaussian-Markov process [Fukunaga, 1990]. Hence, the MFCC features are approximately independent or uncorrelated. Speech recognition systems accumulate second order sufficient statistics to estimate the joint probability density function of the features based on Gaussian distributions. Hence, independent features may lead to the use of Gaussian distributions with diagonal covariance, which may be necessary to have efficient training and recognition algorithms. Perceptual Linear Prediction cepstral coefficient (PLP) features are conceptually similar to MFCCs [Hermansky, 1990]. In this method, the spectrum is warped according to the Bark frequency scale and the logarithmic compression operation is replaced with cube-root compression, which enhances robustness in noisy conditions. The autocorrelation coefficients computed from inverse DCT are modified using autoregressive (all-pole) model to obtain a course representation of the auditory spectrum. Finally, the autoregressive coefficients are converted to cepstral coefficients to obtain a set of independent features.

Since speech is a time varying signal, the basic acoustic features extracted from short time signal analysis do not capture speech dynamics. In order to consider the temporal correlation between the adjacent speech frames, the basic acoustic vector is augmented with its first ($\Delta$) and second order derivatives ($\Delta\Delta$) as dynamic features [Furui, 1986]. These features are usually computed from a window of frames centered around the current frame using a simple regression method. Augmenting the feature vector with the dynamic features leads to significant improvements in recognition performance within the HMM framework.[1] On the other hand, adding the $\Delta$ and $\Delta\Delta$ coefficients to the basic acoustic vector results in the loss of independence of the acoustic vector. This may invalidate the assumptions used for estimating the joint probability density function of the features based on Gaussian distributions. This poor assumption may be relaxed by applying feature projection methods aims to extract (approximately) statistically independent features such as nonlinear Independent Component Analysis (ICA) [Omar and Hasegawa-Johnson, 2003].

The speech signal is speaker dependent and the vocal tract length is one source

[1] The HMM framework will be described in the next section. The use of dynamic features invalidates the conditional independence assumption within the HMM framework. On the other hand, the conventional HMM as a generative model can not generate observations that take into account the inter-frame dependencies imposed by dynamic features. These additional constraints are explicitly modelled within the trajectory HMM framework [Tokuda et al., 2004]. However, these additional constraints are not directly related to the discrimination between speech classes.

of interspeaker variably, where the total length of vocal tract can vary from 13cm for female to 18cm for male. To reduce interspeaker variability, Vocal Tract Length Normalization (VTLN) is a simple method used to warp frequencies in the Mel frequency scale filter bank [Lee and Rose, 1996, Welling et al., 1999]. Based on simple grid search, the optimal VTLN warping factors for each utterance are chosen to maximize likelihoods computed using a speech recognizer.

To increase the environmental robustness of the feature vectors, cepstral mean normalization (CMN) is commonly employed to cancel some channel variations. Since convolutional noise or linear filtering effects in the time domain correspond to a sum in the logarithmic power domain [Gold and Morgan, 1999], the CMN technique is used to subtract the mean of cepstral vectors of the whole utterance ( i.e. an approximation of the channel transfer function) from each (noisy) feature vector that belongs to the same utterance. In the RelAtive SpecTrA (RASTA) approach, the logarithmic spectrum is filtered by a band-pass filter with a sharp spectral zero at the zero frequency of each frequency band [Hermansky and Morgan, 1994]. The high-pass part of the filter is designed to suppress the constant or slowly varying components corresponding to convolutional channel distortions. In addition, the low-pass part of the filter is deigned to smooth the fast frame-to-frame spectral changes. RASTA filtering is similar to a real time implementation of CMN processing. Modulation spectrogram [Kingsbury and Morgan, 1997, Kingsbury et al., 1998] has been designed to handle channel variations.

An ideal feature extraction method for speech recognition would find a set of compact features representing the observation space (i.e. acoustic compression) while preserving the information needed to discriminate between speech classes (i.e. the features are optimal in terms of phone classes separation). This is due to the basic fact that any classification problem is based on two random variables; one representing the observation space and the other representing the classified classes. Hence, a feature extraction method based on cepstral coefficients is not ideal for speech recognition, where it does not take into consideration the discrimination between speech classes or states. Linear Discriminant Analysis (LDA) is a supervised feature extraction method, which finds a linear transform by minimizing the intra-class variance and maximizing the inter-class variance for improved speech recognition [Duda et al., 2000, Haeb-Umbach and Ney, 1992, Hunt et al., 1991, Brown, 1987]. The discriminative coordinates (i.e. basis functions) are determined as the eigenvectors of the ratio of the determinant of the between-class covariance and determinant of the within-class covariance [Duda et al., 2000]. The modelling behind the LDA projection assumes that each class, or state, is modelled by a Gaussian distribution and all the classes share the same covariance matrix. To relax this constraint, variants from LDA such as heteroscedastic linear discriminant analysis (HLDA) [Kumar and Andreou, 1998] or heteroscedastic discriminant analysis (HDA) [Saon et al., 2000] have been formulated as a maximum likelihood estimation problem. These variants have no closed-form solutions and an efficient iterative algorithm was developed to estimate the HLDA projection [Gales, 1999a, Gales, 1999b]. The LDA and its variants are not optimal projection methods in the sense of minimum Bayes classification error. As a result, a linear projection algorithm aims to minimize the probability of misclassification was developed [Saon and Padmanabhan, 2000].

In general, nonlinear discriminant analysis is a more flexible and powerful tool for feature extraction given some additional computational cost. For example, TempoRAl Patterns (TRAPs) and Tandem MLP/HMM approaches extract a feature set computed from longer time intervals than the conventional short-time analysis [Hermansky and Sharma, 1998, Hermansky et al., 2000]. Alternatively, fMPE approach incorporate longer time intervals in a high dimensional space to estimate discriminative non-linearly transformed features using the Minimum Phone Error (MPE) criterion [Povey, 2005, Povey et al., 2005]. Discriminative feature extraction based on optimizing a criterion, which directly measure the recognition error is desirable. Hence, discriminative feature extraction algorithms based on Minimum Classification Error (MCE) criterion [Paliwal et al., 1995, Rahim and Lee, 1996] or MPE criterion [Povey et al., 2005] may be the most objective feature projection algorithms for speech recognition.

Discriminative feature projection algorithms may not change the speech recognition technology based on the HMM framework (i.e. training/decoding); perhaps, this explains why discriminative feature projection is a state-of-the-art approach to improving recognition accuracy and relaxing HMM limitations.

## 2.2 Hidden Markov Model (HMM)

A Hidden Markov Model is a stochastic finite state machine [Rabiner, 1989]. An example of an HMM with left-to-right transition topology, which is used to model a phone in an acoustic model, is shown in figure 2.1. This model has one entry state, three emitting states, and one exit state. The left-to-right topology imposes prior information, where speech production is sequential in time.

For every observation at time $t$, a jump from the current state $i$ to some new state

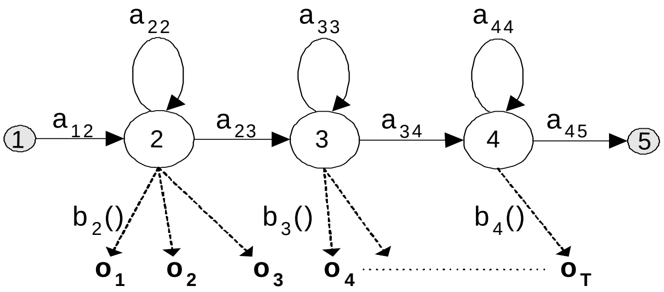


Figure 2.1: A typical Hidden Markov Model for a phone.

$j$ is allowed with a transition probability:

$$a_{ij} = P(\mathbf{s}_{t+1} = j | \mathbf{s}_t = i) \tag{2.2.1}$$

where $\sum_j^N a_{ij} = 1$, N is the number of states in the HMM model. An acoustic feature vector $\mathbf{o}_t$ may be generated, with an output probability density function $b_j(\mathbf{o}_t)$, which is associated with state $j$. A mixture of Gaussian distributions is typically used to model the output distribution for each state,

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}) \tag{2.2.2}$$

where $M$ is the number of mixture components, $c_{jm}$ is the component weight and $\sum_m^M c_{jm} = 1$. $\mu_{jm}$ and $\Sigma_{jm}$ are the component specific mean vector and covariance matrix respectively. If the acoustic features are statistically independent, then diagonal covariance matrices are used to compute the likelihood of a Gaussian model,

$$\mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}) = \prod_{d=1}^{D} \frac{1}{\sqrt{(2\pi)}\sigma_{jmd}} \exp - \frac{(\mathbf{o}_{td} - \mu_{jmd})^2}{2\sigma_{jmd}^2} \tag{2.2.3}$$

where $\sigma_{jmd}$ is the variance element of the Gaussian component $m$ for dimension $d$. An HMM can be written in terms of a set of parameters $\Lambda$,

$$\Lambda = \{a_{ij}, c_{jm}, \mu_{jm}, \Sigma_{jm}\} \tag{2.2.4}$$

HMM model estimation is based on *two* assumptions that lead to a tractable inference when computing the likelihood $p(\mathbf{O}|\mathcal{M})$ of the observation sequence, $\mathbf{O}$, given a model $\mathcal{M}$. Although the HMM is successful as an acoustic model because of these assumptions, they are also its main limitations. The first assumption is the *Markov* assumption, which approximates, or factorizes, the probability of the *hidden* state sequence $\mathbf{S} = \mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_T$ given a model $\mathcal{M}$ by a first order Markov chain:

$$P(\mathbf{S}|\mathcal{M}) = \prod_{t=1}^{T} P(\mathbf{s}_t|\mathbf{s}_{t-1}) = \prod_{t=1}^{T} a_{\mathbf{s}_t\mathbf{s}_{t-1}} \tag{2.2.5}$$

The second assumption is the *conditional independence* assumption, where the probability of an observation sequence, $\mathbf{O}$, given a state sequence, $\mathbf{S}$ and a model $\mathcal{M}$ is given by

$$p(\mathbf{O}|\mathbf{S},\mathcal{M}) = \prod_{t=1}^{T} p(\mathbf{o}_t|\mathbf{s}_t) = \prod_{t=1}^{T} b_{\mathbf{s}_t}(\mathbf{o}_t), \tag{2.2.6}$$

Since the state sequence is hidden, the *total probability* or likelihood of the acoustic observations $p(\mathbf{O}|\mathcal{M})$ is expressed as a sum over all possible state sequences:

$$p(\mathbf{O}|\mathcal{M}) = \sum_{\mathbf{S}} p(\mathbf{O}|\mathbf{S},\mathcal{M})P(\mathbf{S}|\mathcal{M}), \tag{2.2.7}$$

which can be efficiently computed using a dynamic programming algorithm given the factorizations in equation (2.2.5) and equation (2.2.6). The summation over all possible state sequences in equation (2.2.7) can be approximated by a maximum operation to find the best state sequence $\hat{\mathbf{S}}$:

$$\hat{\mathbf{S}} = \arg\max_{\mathbf{S}} p(\mathbf{O}|\mathbf{S},\mathcal{M})P(\mathbf{S}|\mathcal{M}) \tag{2.2.8}$$

which is known as the *Viterbi* path. This gives the best alignment of acoustic observations with the states of an HMM.

### 2.2.1 Generative Parameter Estimation

Parameters of HMMs can be estimated using the maximum likelihood estimate (MLE) framework. For $R$ training observations $\{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_r, \ldots, \mathbf{O}_R\}$ with corresponding transcriptions $\{w_r\}$, the MLE objective function is given by

$$\mathcal{F}_{\text{MLE}}(\Lambda) = \sum_{r=1}^{R} \log p_{\Lambda}(\mathbf{O}_r | \mathcal{M}_{w_r}) \tag{2.2.9}$$

where $\mathcal{M}_{w_r}$ is the composite model corresponding to the reference word sequence $w_r$.

The parameters can be estimated using iterative Baum-Welch algorithm, also known as the *forward-backward* algorithm [Rabiner and Juang, 1993]. The Baum-Welch algorithm is a special case of the Expectation-Maximization (EM) algorithm, which is an efficient iterative procedure to perform MLE in the presence of hidden variables [Dempster et al., 1977]. The inference of an HMM is based on computing the forward and backward probabilities. The forward probabilities can be computed recursively:

$$\alpha_j(t) = p(\mathbf{o}_1, \ldots, \mathbf{o}_t, \mathbf{s}_t = j | \mathcal{M}) = \Big( \sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} \Big) b_j(\mathbf{o}_t) \tag{2.2.10}$$

with initial conditions $\alpha_1(1) = 1$ and $\alpha_j(1) = a_{1j} b_j(\mathbf{o}_1)$ for $1 < j < N$ and a final condition $\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}$. Similarly, the backward probabilities can be computed:

$$\beta_j(t) = p(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T | \mathbf{s}_t = j, \mathcal{M}) = \sum_{i=2}^{N-1} a_{ji} b_j(\mathbf{o}_{t+1}) \beta_i(t+1) \tag{2.2.11}$$

with initial conditions $\beta_i(T) = a_{iN}$ for $1 < i < N$ and a final condition $\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\mathbf{o}_1) \beta_j(1)$.

The frame-state alignment probability $\gamma_j$, denoting the probability of being in state $j$ at some time $t$ can be written in terms of the forward probability $\alpha_j(t)$ and the backward probability $\beta_j(t)$:

$$\gamma_j(t) = P(\mathbf{s}_t = j|\mathbf{O};\mathcal{M}) = \frac{p(\mathbf{O},\mathbf{s}_t = j|\mathcal{M})}{p(\mathbf{O}|\mathcal{M})} = \frac{\alpha_j(t)\beta_j(t)}{p(\mathbf{O}|\mathcal{M})} \tag{2.2.12}$$

where

$$p(\mathbf{O}|\mathcal{M}) = \alpha_N(T) = \beta_1(1) \tag{2.2.13}$$

and a component specific alignment probability can be derived:

$$\gamma_{jm}(t) = P(\mathbf{s}_t = j, \mathbf{m}_t = m|\mathbf{O};\mathcal{M}) = \gamma_j(t)\frac{c_{jm}b_{jm}(\mathbf{o}_t)}{b_j(\mathbf{o}_t)} \tag{2.2.14}$$

where the $m^{th}$ component is associated with the $j^{th}$ state.

Consequently, the accumulators of the sufficient statistics $\mathcal{C}_{jm}(1) = \gamma_{jm}$, $\mathcal{C}_{jm}(\mathbf{O})$, and $\mathcal{C}_{jm}(\mathbf{O}^2)$ are calculated as follows:

$$\begin{aligned}
\mathcal{C}_{jm}(1) &= \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{jm}^{r}(t) && (2.2.15)\\
\mathcal{C}_{jm}(\mathbf{O}) &= \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{jm}^{r}(t)\mathbf{o}_t && (2.2.16)\\
\mathcal{C}_{jm}(\mathbf{O}^2) &= \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{jm}^{r}(t)\mathbf{o}_t^2 && (2.2.17)
\end{aligned}$$

Hence, the Baum-Welch re-estimation formulae for the mean and covariance of state $j$ and component $m$ of an HMM are given by

$$\begin{aligned}
c_{jm} &= \frac{\mathcal{C}_{jm}(1)}{\sum_m \mathcal{C}_{jm}(1)} && (2.2.18)\\
\mu_{jm} &= \frac{\mathcal{C}_{jm}(\mathbf{O})}{\mathcal{C}_{jm}(1)} && (2.2.19)\\
\Sigma_{jm} &= \frac{\mathcal{C}_{jm}(\mathbf{O}^2)}{\mathcal{C}_{jm}(1)} - \mu_{jm}^2 && (2.2.20)
\end{aligned}$$

The transition probabilities between states are also estimated by calculating the forward and backward probabilities.

Generative training of HMM models leads to models that may be useful for generating speech, which is useful for speech synthesis. Using Bayes rule as in equation (1.1.1), these generative models can be used for speech recognition. Although HMM models should be trained to discriminate between speech classes, it is not uncommon that generative training is the basic training method in speech recognition. This is related to the basic fact that generative training of HMM models is fast and efficient because:

- The Maximization step of the EM algorithm for Gaussian models inherits the closed form of Gaussian's mean and variance estimation from the data. This attractive property may be the main reason behind the widespread of HMM models.

- Maximum likelihood generative training accumulates statistics from the correct class only (i.e. it does not use out-of-class data for discrimination). This leads to a fast training for speech recognition, where the data is split according to classes and trained independently. However, this advantage is a common property of generative training.

- Controlling the number of Gaussian mixtures usually leads to coarse generative modelling, which is usually very effective for modelling the spectral information related to discrimination. Modelling the fine structure of the spectrum may lead to poor discrimination.

HMMs trained by a generative training procedure maximize the likelihood between the data and the underlying distributions. However, if the true underlying distribution that generated the data is an HMM, given sufficient data, the Bayes classification based on the HMM models, will minimize the probability of classification/recognition error [Nadas, 1983]. Practically, the decision boundaries constructed after the generative training are not optimal and generative HMMs are not optimal models for speech recognition applications. One way to address this problem within the HMM framework is to utilize the parameters efficiently to improve the discrimination between speech classes via discriminative training for HMM models [Bahl et al., 1986, Brown, 1987].

### 2.2.2 Discriminative Parameter Estimation

HMM models trained using the EM algorithm are very effective for coarse generation of data. Unfortunately, generative training does not address the classification problem, where the objective is to discriminate between the classes and hence to reduce the misclassification error. To address this problem, the Gaussians of an HMM can be rotated and shifted in the feature space to increase the discrimination between classes via a discriminative training procedure.

The Conditional Maximum Likelihood (CML) criterion, defined by equation (2.2.21), aims to maximize the log of posterior probability of the correct word sequence given the observations,

$$\begin{aligned}\mathcal{F}_{\text{CML}}(\Lambda) &= \sum_{r=1}^{R} \log P_\Lambda(\mathcal{M}_{w_r}|\mathbf{O}_r) \\ &= \sum_{r=1}^{R} \log \frac{p_\Lambda(\mathbf{O}_r|\mathcal{M}_{w_r})P(w_r)}{\sum_{\hat{w}} p_\Lambda(\mathbf{O}_r|\mathcal{M}_{\hat{w}})P(\hat{w})} \\ &\approx \sum_{r=1}^{R} \log p_\Lambda(\mathbf{O}_r|\mathcal{M}_r^{\text{num}}) - \log p_\Lambda(\mathbf{O}_r|\mathcal{M}_r^{\text{den}})\end{aligned} \tag{2.2.21}$$

where $\mathcal{M}_w$ is a composite model corresponding to the word sequence $w$ and $P(w)$ is the probability of this sequence as determined by a language model. This discriminative training aims to maximize a term related to the probability of the correct models (known as the numerator) $p_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{num}})$, which is identical to the ML objective function, and simultaneously minimize a term related to all models probabilities (known as the denominator term) $p_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{den}}) \approx \sum_{\hat{w}} p_\Lambda(\mathbf{O}_r|\mathcal{M}_{\hat{w}})P(\hat{w})$. $\sum_{\hat{w}} p_\Lambda(\mathbf{O}_r|\mathcal{M}_{\hat{w}})P(\hat{w})$, which is the summation over all possible word sequences $\hat{w}$ allowed in the task, is computationally expensive for LVCSR systems. As a result, $p_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{den}})$ is an approximation to the denominator term, which is computed by N-best lists [Chow, 1990] or lattices [Normandin et al., 1994, Valtchev et al., 1997] generated from a decoding pass based on MLE trained models.

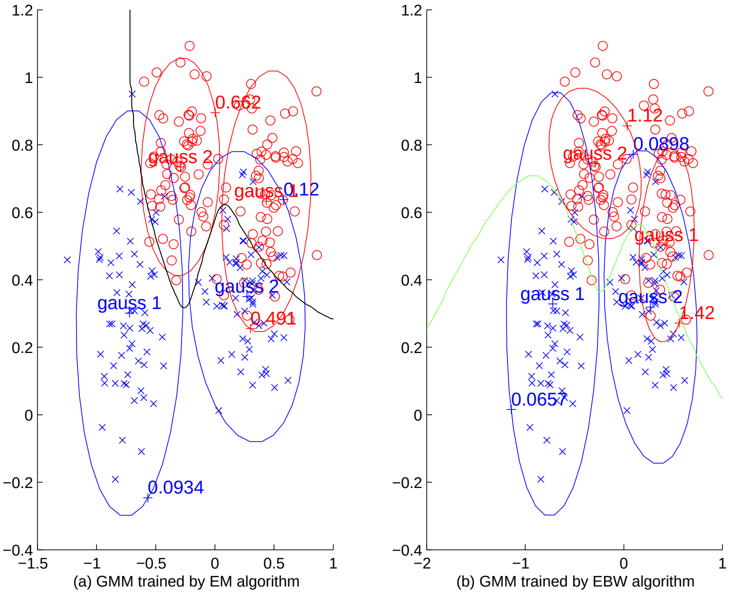
Extended Baum-Welch (EBW) algorithm is the state-of-the-art discriminative training algorithm that maximize the CML criterion for HMMs.[2] It was introduced for discriminative training for discrete distributions in [Gopalakrishnan et al., 1991]. Using a discrete approximation to the Gaussian distribution [Normandin, 1991], it was shown that the mean of a particular dimension of the Gaussian for state $j$, mixture component $m$, $\mu_{jm}$ and the corresponding variance, $\sigma^2_{jm}$ (assuming diagonal covariance matrices) can be reestimated by

$$\hat{\mu}_{jm} = \frac{\mathcal{C}^{\text{num}}_{jm}(\mathbf{O}) - \mathcal{C}^{\text{den}}_{jm}(\mathbf{O}) + D\mu_{jm}}{\gamma^{\text{num}}_{jm} - \gamma^{\text{den}}_{jm} + D} \tag{2.2.22}$$

$$\hat{\sigma}^2_{jm} = \frac{\mathcal{C}^{\text{num}}_{jm}(\mathbf{O}^2) - \mathcal{C}^{\text{den}}_{jm}(\mathbf{O}^2) + D(\mu^2_{jm} + \sigma^2_{jm})}{\gamma^{\text{num}}_{jm} - \gamma^{\text{den}}_{jm} + D} \tag{2.2.23}$$

In these equations, $D$ is a smoothing constant that controls the degree of deviation of the new parameters with respect to the old parameters. The superscripts *num* and

[2] For numerical optimization based methods see [Kapadia et al., 1993, Kapadia, 1998]. Given an appropriate setting for learning parameters and smoothing terms, the EBW and gradient ascent algorithms can be equivalent [Schlüter et al., 1997].



(a) GMM trained by EM algorithm

(b) GMM trained by EBW algorithm

Figure 2.2: Two class classification problem. (a) decision boundary is constructed with EM generative training (b) decision boundary is constructed by EBW discriminative training.

*den* refer to the model corresponding to the correct word sequence, and the recognition model for all word sequences, respectively. Figure 2.2 shows the decision boundary for a simple two class classification problem, where the Gaussians are shifted and rotated to improve the discrimination between classes. This may explain the basic idea behind the EBW update for Gaussian models. It may be important to mention that Gaussian models estimated by discriminative training are not generative models. They are simply activation functions that have the same functional form of a Gaussian generative model and the same probabilistic constraint (i.e. $\int_{\mathbf{o}} f(\mathbf{o}|\mu, \Sigma)d\mathbf{o} = 1$). Similarly, HMM models trained using discriminative procedures are not generative models or distributions.

Setting the optimal value for $D$ is the subject of extensive research and it is usually set per-Gaussian level, $D_{jm}$, given the formula

$$D_{jm} = \max\{2D_{jm}^{min}, E\gamma_{jm}^{den}\}, \tag{2.2.24}$$

where $D_{jm}^{min}$ is a necessary value to ensure positive variances and $E$ is a global constant set to 1 or 2 [Woodland and Povey, 2000]. It has been shown that there is a value of $D$, which proves the convergence of the algorithm [Gunawardana and Byrne, 2001] and [Kanevsky, 2004]. Using the reverse Jensen inequality for e-family distributions [Jebara, 2002], a closed form expression for $D_{jm}$ was derived and the heuristic in equation (2.2.24) was justified [Afify, 2005]. The discriminative training of HMM models is usually initialized by ML generative training. For historical reasons, CML discriminative training for HMMs is known as Maximum Mutual Information Estimation (MMIE) in speech recognition domain. The two criteria lead to the same results because the language model parameters are not optimized during the training.

Discriminative training based on the CML objective function does not directly measure the expected WER criterion. Instead, the Overall Risk Criterion Estimation (ORCE) [Na et al., 1995] directly minimizes the expected word or phone error rates by refining the model parameters based on a measure of risk related to recognition error. The update equations of the parameters for ORCE was shown to be very similar to the EBW update equations described above for CML [Kaiser et al., 2002]. Minimum Phone Error (MPE) criterion may be considered as a particular realization of ORCE and it is given by

$$\mathcal{F}_{\mathrm{MPE}}(\Lambda) = \sum_{r=1}^{R} \sum_{w} P_{\Lambda}(\mathcal{M}_w|\mathbf{O}_r) A(w, w_r) \tag{2.2.25}$$

where $A(w, w_r)$ is the raw phone transcription accuracy of the sentence $w$, given the reference sentence $w_r$. It has been reported that ORCE based on MPE criterion gives a small improvement over ORCE based on Minimum Word Error (MWE) criterion [Povey and Woodland, 2002, Povey, 2004]. Alternatively, the Minimum Classification Error (MCE) criterion [Juang and Katagiri, 1992] may be used to update the parameters of HMMs [Chou et al., 1992, Chou et al., 1993, Reichl and Ruske, 1995, Katagiri et al., 1998]. However, most typical MCE implementations measure sentence error rate and thus do not directly measure the WER. Several methods to refine the MCE approach have been proposed. These methods are based on updating the parameters of HMMs based on large margin estimation or its variants [Li et al., 2005, Liu et al., 2005] and soft margin estimation (SME) [Li et al., 2006]. Margin estimation methods aim to maximize the separation (margin) between two classes (see Chapter 4). The SME criterion may be more effective than the MCE criterion because the former takes into account the misclassified examples, which are far from the decision boundary during the training process [Li et al., 2006]. Some discriminative criteria have been compared in a unified framework for some tasks [Macherey et al., 2005, Schlüter et al., 2001]. To match the training and decoding criteria, ORCE based criteria can also be used for decoding tasks since they directly minimize the expected word error rate [Mangu et al., 1999, Mangu et al., 2000].

## 2.3 Beyond HMM

Large vocabulary continuous speech recognition systems based on continuous Gaussian mixture HMMs are very successful [Young, 1996]. However, the HMM has been criticized and some alternatives have been suggested to overcome its known limitations. For example, the inherited state duration distribution of HMM state is a geometric distribution, which cannot represent a phone duration [Rabiner, 1989].

Hence, explicit duration models based on the gamma distribution have been introduced within the HMM framework [Levinson, 1986]. It has been argued that first order Markov chain which helps to factorize the joint probability of the *hidden* state sequence is a limitation but that it was shown that any $n^{th}$ Markov chain may be transformed into a first order Markov chain [Jelinek, 1997, Bilmes, 2006]. The conditional independence assumption has been advocated [Tóth and Kocsor, 2005] based on that naive Bayes provides optimal classification even though it incorrectly estimates the probabilities [Domingos and Pazzani, 1997], where the wining class probability dominate other class probabilities (i.e. the model is too confident in its decision).[3] In "What HMMs can do" [Bilmes, 2006], it appears that there is very little an HMM cannot do and that the problem may be related to how HMMs are used.

Recognition accuracy can be significantly increased by increasing the number of hidden states in the HMM. We refer to this process as *augmenting the state space*, which aims to increase the capacity of observation distributions. This is usually done by using context-dependent HMM models like tri-phone, quad-phone, or penta-phones, which use a window of left and right neighboring phones. The process of augmenting the state space increases dramatically the number of parameters, which need to be robustly estimated given the limited amount of training data and unseen context. Parameter tying allows acoustically similar units to share the same parameters. Extensive research has been done on clustering the augmented state space based on tied, context-dependent phonetic units to reduce model complexity given limited training data [Lee, 1988, Hwang and Huang, 1991, Bahl et al., 1991,

[3] In other words, it does not matter if the classifier scores are accurate or not but it matters that entropy of the classifier scores is low (i.e. sharp posteriors). As a result, the classifier can produce confident decisions.

Young and Woodland, 1994].

Buried Markov models (BMM) [Bilmes, 1998, Bilmes, 1999] extend the HMM by integrating specific cross-observation dependencies to relax HMM conditional independence assumptions. Conditioning the likelihood of the current observation on a random variable representing the acoustic context information $\mathbf{z}_t$ related to the current observation $\mathbf{o}_t$ (i.e. $p^{HMM}(\mathbf{o}_t|\mathbf{s}_j) \Rightarrow p^{BMM}(\mathbf{o}_t|\mathbf{s}_j, \mathbf{z}_t)$) is the main idea behind this work. For each $\mathbf{s}_j$, element $i$ in acoustic vector $\mathbf{o}_t$, and $c$ previous frames, the *additional dependencies* between observations elements $\mathbf{z}_{ti} = \{\mathbf{o}_{t-c,k} : \forall c, k \leq d\} - \mathbf{o}_{ti}$ are computed. The *relevant* additional dependencies $\mathbf{z}_{si} \in \mathbf{z}_{ti}$ with an observation element $\mathbf{o}_{ti}$, given a state $\mathbf{s}_j$, are derived or selected directly from the training corpus based on a discriminative entropy reduction measure. Hence, the BMM model is trained using an EM algorithm, which estimates data driven, sparse matrices determined by the BMM dependencies for each state $\mathbf{s}_j$. This may lead to a data driven sparse acoustic context modelling in the original feature space within the HMM framework. Although our work uses another modelling approach and mathematical tools, our technique, based on $l_1$-ACRF modelling (see Chapter 6), is conceptually similar to BMM modelling, where the aim is to take advantage of the acoustic context to relax the HMM conditional independence assumption. The major difference between BMM and $l_1$-ACRF modelling is that we learn the sparse structure of the model and the parameters concurrently in a principled way and the compression of acoustic context information is done in the augmented spaces rather than in the original feature space. Augmented spaces are more likely to be linear separable and integrating the acoustic context information may be more effective in the augmented spaces. BMMs were not motivated in a discriminative modelling framework and are trained using generative training but the additional cross-observation information was selected based on discriminative criterion.

In hybrid HMM/ANN speech recognition systems [Morgan and Bourlard, 1995], Artificial Neural Networks (ANN) models are used as flexible discriminant classifiers to estimate a scaled likelihood. In particular, the emission probability score is given by

$$b_j(\mathbf{o}_t) = \frac{P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)}{P(\mathbf{s}_j)} \tag{2.3.1}$$

where $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is the posterior probability of a phonetic state estimated by a connectionist estimator [Trentin and Gori, 2001] and $P(\mathbf{s}_j)$ is estimated from the labelled data. In addition to discriminative training, if the posterior probability $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is sensitive to acoustic context, $b_j(\mathbf{o}_t)$ score may help to overcome conditional independence assumption and improve the overall recognition performance without changing the basic HMM framework.

Recently, DBN based acoustic modelling were developed [Zweig and Russell, 1998, Bilmes and Zweig, 2002, Bilmes and Bartels, 2005]. Basic introduction to graphical modelling is given in Chapter 1. Unlike HMMs, DBNs offer a flexible statistical framework for developing a large number of acoustic models or even complete speech recognition systems in an easy and principled way. Hybrid approaches based on ANN/DBN [Frankel and King, 2005] or SVM/DBN [Hasegawa-Johnson et al., 2005] were developed for flexible information fusion. Ongoing research on DBN models may lead to a model that outperforms the HMM but there is no evidence, as yet, to support this.

Beyond frame-based modelling like HMM, segment or trajectory models have been developed [Ostendorf et al., 1996]. Unlike HMMs, these models consider the phone as a segment rather than the product of frame level probabilities. Segmental models are motivated based on a basic fact that the speech signal is produced by a continuous, time varying, physical system. Hence, the HMM conditional independence assumption, which results in a piece-wise stationary process within an HMM state is inadequate to model speech process. Each state in a segmental model generates a variable length observation sequence. The output distribution given a discrete state $\mathbf{s}$ and an observation sequence of length $L$ is given by

$$p(\mathbf{o}_1^L|\mathbf{s},\mathcal{M}) = p(\mathbf{o}_1^L|\mathbf{s},L)P(L|\mathbf{s}) \tag{2.3.2}$$

Segmental models can be viewed as a variant of HMMs, which have an explicit state duration modelling and the state distribution is a function of the entire sequence of frames of that state. Thus, the segmental models aim to capture speech dynamic behavior over a longer time interval. However, segmental models have lead to modest improvements with a considerable increase of complexity with respect to HMMs. In the near future, it is not likely that segmental models will replace HMMs or simple DBNs given their excellent computational properties. This thesis is developed within the frame-based modelling. Hence, segmental models are outside the scope of this research. Further details about segmental models can be found in [Digalakis, 1992, Holmes and Russell, 1999, Richards and Bridle, 1999, Deng et al., 1994, Gales and Young, 1993, Frankel and King, 2007].

## 2.4 Speech Recognition Committees

A complex computational task like speech recognition may be solved by breaking down it into a large number of computationally simple tasks. The simple tasks or *experts* can focus on one or more modelling aspects to improve the overall speech recognition performance. For example, speech recognition experts can be trained on different set of acoustic features since every feature set has its own strengths and weaknesses. The highly specialised experts are then combined given an optional combination strategy. The combination between experts is said to constitute a *committee machine* [Haykin, 1998]. The knowledge acquired by a committee of experts is supposedly superior to that gained by any individual expert. In general, committee machine methods can reduce the computational cost associated with very complex speech recognition systems. In addition, these methods may improve acoustic modelling resolution, robustness in noisy environments, and the overall speech recognition performance.

In general, the combination of experts for speech recognition can be performed for different goals. For example, it was shown that the accuracy of a large vocabulary speech recognition system can be increased by combining the final hypotheses of individual recognizers using Recognizer Output Voting Error Reduction (ROVER) [Fiscus, 1997]. This combination works at the highest level of speech recognition systems. A system combining multiple classifiers trained to enriching the acoustic modelling from different heterogeneous acoustic measurements chosen to improve the recognition performance was developed [Halberstadt and Glass, 1998]. A similar approach based on combining a large number of classifiers was developed for landmark speech recognition [Hasegawa-Johnson et al., 2005]. Multi-band speech recognition is based on the recombination of speech recognizers or experts, each one working in a specific frequency band [Okawa et al., 1998, Bourlard and Dupont, 1996, Hagen et al., 1998, Hagen and Bourlard, 2001]. In addition, combining articulatory and acoustic information may be helpful in noisy and reverberant environments [Kirchhoff, 1998]. Hence, the combination in adverse conditions aims to improve the robustness of speech recognition systems for noise conditions.

A general approach for discriminative model combination of many acoustic models and language models was introduced [Beyerlein, 1998]. This approach is based on a linear combination model of different experts. The parameters of the combination model are estimated based on MCE training to minimize the empirical error rate on training data. This approach may be the most objective combination strategy since it directly measures the recognition performance. This advantage comes with the additional cost of estimating the combination model parameters. Although the algorithm was developed to combine higher level speech recognition components, a similar idea may be developed for other goals such as optimal combination in multi-band speech recognition [Cerisara et al., 1999]. On the other hand, information theoretic measures such as maximum mutual information may be used to design simple combination strategies [Okawa et al., 1999, Ellis and Bilmes, 2000, Ellis, 2000]. In addition, a combination strategy based on linear transforms such as LDA and Principle Component Analysis (PCA) are used to extract acoustic features to improve speech recognition [Morgan et al., 2005, Hermansky and Sharma, 1998, Hermansky et al., 2000].

# Chapter 3

# The Maximum Entropy Principle

## 3.1 Introduction

The Maximum Entropy (MaxEnt) principle is a method used to select a probability distribution for a given stochastic system with known *states* or outputs. This selected probability distribution should be an optimum in some sense. The collected prior knowledge about the system is available in the form of *constraints*. Since the available information about the system is incomplete, there is an infinite number of possible distributions that satisfy the constraints on the probability distribution over the states. Which distribution should be selected?

The answer to this fundamental problem lies in maximizing Shannon's entropy as suggested by E. T. Jaynes [Jaynes, 1957]. The maximum entropy principle is stated by Jaynes [Jaynes, 1982]:

> When we make inferences based on incomplete information, we should draw them from that probability distribution that has the maximum entropy permitted by the information we do have.

The MaxEnt method encourages us to choose the distribution that has maximum uncertainty, randomness, is most unbiased or uniform, and is simultaneously consistent with mean values of the constraints. It is a natural extension of Laplace's principle of sufficient reason, which postulates that "*in the absence of any constraint, all outcomes are equally likely*" [Kapur and Kesavan, 1992]. This means that the uniform distribution is the most intuitive distribution in the absence of prior knowledge.

Table 3.1: *The MaxEnt solution for a fair die (no constraints) is a uniform distribution.*

| $\mathbf{s}$ | **1** | **2** | **3** | **4** | **5** | **6** | $S(\mathbf{s})$ |
|---|---|---|---|---|---|---|---|
| $P(\mathbf{s})$ | 1/6 | 1/6 | 1/6 | 1/6 | 1/6 | 1/6 | 1.7918 |

For example, when we throw a die, we assign equal probability to each face as shown in Table 3.1. Imagine an unfair die, where face one has probability $P(1) = 0.15$ and face five has probability $P(5) = 0.05$, what is the MaxEnt distribution representing this unfair die? To formulate this problem, equations (3.1.1), (3.1.2), and (3.1.3), represent the three prior constraints. Obviously, we have infinite numbers of solutions for this problem. For example, Table 3.2 is an optional distribution (solution) with the same constraints, but this solution does not maximize the uncertainty in terms of Shannon's entropy measure and it is not the maximum entropy solution. The MaxEnt solution is shown in Table 3.3 and equation (3.1.4) represents the unfair die distribution, where $\mathbf{s}$ is a discrete random variable representing each face of the die and $f_1(\mathbf{s})$, $f_5(\mathbf{s})$ are binary constraints. $f_1(\mathbf{s}) = 1$ for face one only and $f_5(\mathbf{s}) = 1$ for face five only. This solution considers *only* the collected prior knowledge about the die.

$$P(1) + P(2) + P(3) + P(4) + P(5) + P(6) = 1.0 \tag{3.1.1}$$

$$P(1) = 0.15 \tag{3.1.2}$$

$$P(5) = 0.05 \tag{3.1.3}$$

$$P_\Lambda(\mathbf{s}) = \frac{1}{5}\exp(-0.287682 f_1(\mathbf{s}) - 1.38629 f_5(\mathbf{s})) \tag{3.1.4}$$

Table 3.2: *An optional solution that satisfies the given constraints but it is not the MaxEnt solution.*

| s | **1** | **2** | **3** | **4** | **5** | **6** | $S(\mathbf{s})$ |
|---|---|---|---|---|---|---|---|
| $P(\mathbf{s})$ | 0.15 | 0.2 | 0.2 | 0.3 | 0.05 | 0.1 | 1.4727 |

Table 3.3: *MaxEnt solution for unfair die.*

| s | **1** | **2** | **3** | **4** | **5** | **6** | $S(\mathbf{s})$ |
|---|---|---|---|---|---|---|---|
| $P(\mathbf{s})$ | 0.15 | 0.2 | 0.2 | 0.2 | 0.05 | 0.2 | 1.7219 |

In general, for simple problems we can obtain the MaxEnt solution intuitively. However, for large problems and large numbers of constraints, we should have a clear mathematical treatment for the MaxEnt problem. This is the subject of the following section.

Most of the theoretical probability distributions that have appeared in the literature can be derived on the basis of the MaxEnt principle when the values of one or more of the constraints are changed. For more details, the reader can refer to [Kapur and Kesavan, 1992, page 65-69].

## 3.2 The Maximum Entropy Principle Formalism

Let **s** be a discrete variable representing the possible output classes/states in a classification problem, and **o** be an observation affecting the states of the system. The constrained optimization problem at hand is to maximize the conditional Shannon entropy:

$$\arg\max_{P \in \mathcal{C}} S(P) = -\sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \ln P_\Lambda(\mathbf{s} \mid \mathbf{o}) \tag{3.2.1}$$

subject to

C1 $P_\Lambda(\mathbf{s} \mid \mathbf{o}) \geq 0$ for all $\mathbf{s}$ and $\mathbf{o}$.

C2 $\sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) = 1$ for all $\mathbf{o}$.

C3 $\sum_{\mathbf{o}} p(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o}, \mathbf{s}) = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}, \mathbf{s}) g_i(\mathbf{o}, \mathbf{s}) = \tilde{p}(g_i)$ for $i = 1, 2, \ldots, n$.

Where $S(P)$ is the expectation of the conditional entropy of the model with respect to the training data, $\tilde{p}(\mathbf{o})$ is the observed marginal probability, and $\Lambda = \{\lambda_i\}$ is the set of parameters to be optimized. Constraints C1 and C2 represent the direct constraints from probability theory. Constraint C3 represents the integration of the available prior knowledge on the random variables $\mathbf{o}$ and $\mathbf{s}$ in terms of the characterizing constraints $g_i(\mathbf{o}, \mathbf{s}) = \delta(\mathbf{s}, \mathbf{s}') g_i(\mathbf{o})$, which have expected values $\tilde{p}(g_i)$. The empirical knowledge is collected via the noisy training data, $\mathcal{D} = \{(\mathbf{o}_t, \mathbf{s}_t)\}_{t=1}^{T}$ and there is no information available about the form of the function that defines the relationship between $\mathbf{o}$ and $\mathbf{s}$.

The maximum entropy[1] formalism results in a probability distribution, which is the log linear or exponential model:

$$P_\Lambda(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_\Lambda(\mathbf{o})} \exp\Big(\sum_i \lambda_i g_i(\mathbf{o}, \mathbf{s})\Big) \tag{3.2.2}$$

where

- $\lambda_i$ is the Lagrange multiplier (weighting factor) associated to the function $g_i(\mathbf{o}, \mathbf{s})$.

[1] Maximizing the Shannon's entropy subject to the constraints is equivalent to minimizing the KL divergence of $P$ from the uniform distribution $Q = 1/|\mathbf{s}|$ subject to the same constraints (see Chapter 1).

- $Z_\Lambda(\mathbf{o})$ (Zustandsumme) is a normalization coefficient resulting from the natural constraints over the probabilities summation, commonly called the partition function, and given by

$$Z_\Lambda(\mathbf{o}) = \sum_{\mathbf{s}} \exp\Big(\sum_i \lambda_i g_i(\mathbf{o}, \mathbf{s})\Big)$$

Conditional exponential models in equation (3.2.2) implies undirected graphical models or conditional random fields when the potentials are exponential functions (see Chapter 1). As a result, we refer to any conditional exponential model (MaxEnt) model as a CRF model in this thesis.

The entropy is a concave function of the mean values of the characterizing constraints $\tilde{p}(g_i)$ [Kapur and Kesavan, 1992]. Hence, the MaxEnt solution is unique given the empirical mean values of the constraints. As a result, the solution is not sensitive to the initial values of the model parameters and the constructed model is unique for a given database in the statistical learning procedure. In contrast, flexible constraint formulation may change this property as detailed in the next section.

It should be noted that in the absence of any constraint, other than the natural constraint, the maximum entropy formalism results in the uniform distribution:

$$P_\Lambda(\mathbf{s} \mid \mathbf{o}) = 1/|\mathbf{s}|. \tag{3.2.3}$$

This result explains the basic philosophy behind maximizing the entropy as the uniform distribution produces the most unbiased distribution. Integrating constraints results in reduction of the entropy but the output distribution is the most unbiased distribution consistent with constraints.

Consider a maximum entropy problem with two constraints $\mu$ and $\sigma^2$ of a continuous random variable whose probability density function is square-integrable. In such a case, when the continuous entropy is maximized, its solution is the normal distribution. This explains the importance of this distribution and why it has been frequently used in the application of statistical inference and why it deserves the adjective *normal*, where this distribution is the most uncertain and maximizes the entropy [Guiasu and Shenitzer, 1985].

## 3.3 Constraint Formulation

According to the MaxEnt principle, the prior knowledge is represented by the characterizing moment functions $g_i(\mathbf{o}, \mathbf{s}) = \delta(\mathbf{s}, \mathbf{s}')g_i(\mathbf{o})$. The characterizing moment functions or constraints can be described as an optional implementation issue. Choosing different $g_i(\mathbf{o})$ will lead to different MaxEnt or CRF classifiers. Table 3.4 summaries the most common choices of the characterizing moment functions $g_i(\mathbf{o})$ and their corresponding CRF classifiers.

Table 3.4: *Common MaxEnt characterizing moment functions.*

| $g_i(\mathbf{o})$ | name | problem dimensionality | CRF Classifier |
|---|---|---|---|
| 1 | bias | - | used in all models |
| $\mathbf{o}$ | $1^{\text{st}}$ order moments | low | linear logistic regression |
| $\mathbf{o}^2$ | $2^{\text{nd}}$ order moments | low | quadratic logistic regression |
| $\mathcal{N}(\mathbf{o}; \mu, \Sigma)$ | radial basis | medium | probabilistic ANNs |
| $\frac{1}{1+\exp(\lambda^T \mathbf{o})}$ | sigmoid | medium | probabilistic ANNs |
| $k(\mathbf{o}, \acute{\mathbf{o}})$ | kernel | high | kernel logistic regression |

The constraints $g_i(\mathbf{o})$ based on low dimensional spaces like $\mathbf{o}, \mathbf{o}^2$ are useful to construct linear or quadric discriminant classifiers. Such models appear as logistic regression in statistical learning [Hastie et al., 2001]. Higher order moments can lead to the construction of general polynomial CRF classifiers but it is not common as there are more efficient methods to construct nonlinear classifiers.

On the other hand, the constraints $g_i(\mathbf{o})$ based on high dimensional spaces is usually achieved by mapping the low dimensional input space into a high dimensional space and construct a linear decision boundaries with the MaxEnt principle. The back propagation neural networks with softmax output layer construct such spaces by using a large number of sigmoid activation functions [Bridle, 1990a, Bridle, 1990b]. The radial basis neural networks with softmax layer construct such spaces by clustering the data into large number of Gaussians representing the dense regions of the observation space. Both CRF classifiers can be understood as probabilistic ANNs and the solution is not unique. The formal solution of the minimization of the Mean Square Error (MSE) as a measure of dispersion of distributions $\sum(\tilde{P} - P_\Lambda)^2$, given the constraints, will lead to linear probability distributions

$$P_\Lambda(\mathbf{s} \mid \mathbf{o}) = \sum_i \lambda_i g_i(\mathbf{o}, \mathbf{s}) \tag{3.3.1}$$

which lacks the property of nonnegativity [Jaynes, 2003]. Hence, radial basis neural networks trained using quadratic optimization cannot have probabilistic interpretation.

Kernel methods construct high dimensional spaces by mapping the low dimensional input space into some other dot product space (i.e. feature space) via a nonlinear transformation. The state-of-the-art nonparametric classification algorithm, Support Vector Machine (SVM) is formulated based on such spaces [Vapnik, 1998]. The feature spaces dimensionality is related to the number of training data points $T$ and the classification algorithm will focus on selecting the most useful data points representing the whole training data. This implies compression of the observation space, while maintaining the discrimination capability with a certain accuracy. The import vector machine is an implementation of kernel logistic regression [Zhu and Hastie, 2001]. In Chapter 4, an efficient implementation of kernel logistic regression is also addressed, which is suitable for large scale classification.

The classifiers based on equation (3.2.2) are suitable for static pattern classification. Sequential CRF classifiers based on the MaxEnt principle are also formulated [Lafferty et al., 2001]. As with static classifiers, alternative definitions of $g_i(\mathbf{o})$ can lead to different sequential classifiers. Sequential CRFs based on kernel spaces are addressed in [Lafferty et al., 2004]. Sequential CRFs for speech recognition are addressed in Chapter 5 and Chapter 6.

## 3.4 Parameter Optimization

Parameter Optimization for MaxEnt models has no analytical solutions. Instead, the Lagrange multipliers, $\Lambda$, are estimated using numerical methods.

Section 3.4.1 will review the relationship between two different frameworks based on the MaxEnt and maximum likelihood principles. As a result, the maximum likelihood parameter estimation for the MaxEnt models is a direct application for this relationship. In Section 3.4.2, the maximum likelihood parameter estimation for the MaxEnt models based on general purpose nonlinear numerical optimization is described. Alternatively, lower bound parameter optimization is described in Section 3.4.3. Lower bound optimizers take advantage of the fact that conditional maximum likelihood (CML) estimation of MaxEnt models has a special structure. Hence, by optimizing a lower bound of the true CML objective function, it is easy to find a *simple principled* update to the parameters iteratively.

The MaxEnt solution can be either a global maximum or a local maximum, which depends on how the constraints are formulated. According to our definition of the constraints in Section 3.3, probabilistic ANNs formulation will lead to a local maximum since there are many ways to specify the activation functions and the other forms will lead to global solutions.

### 3.4.1 Maximum Entropy and Maximum Likelihood

Maximum likelihood parameter estimation for exponential models and MaxEnt parameter estimation are the same even though they are two different approaches for statistical inference [Della Pietra et al., 1997]. An intuitive interpretation for this fact is that the MaxEnt method selects an exponential distribution that maximizes the entropy. Once the distribution is known, the maximum likelihood method will fit the exponential distribution to the data. On the other hand, MaxEnt will not assume anything more than the mean values of moments that define the constraints collected from the data. Thus, to minimize the entropy of the sample (empirical) data, we have to maximize the likelihood function $\Lambda^* = \arg\max_\Lambda \mathcal{F}_{\text{CML}}(\Lambda)$, which yields the optimal parameters $\Lambda^*$. The log likelihood function of the exponential model in equation (3.2.2) with respect to the empirical distribution $\tilde{p}$ is given by

$$\mathcal{F}_{\text{CML}}(\Lambda) = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \lambda_i g_i(\mathbf{o},\mathbf{s}) - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log \sum_{\mathbf{s}} \exp \sum_i \lambda_i g_i(\mathbf{o},\mathbf{s}) \tag{3.4.1}$$

and $\Lambda^*$ is obtained when

$$\frac{\partial \mathcal{F}_{\text{CML}}(\Lambda)}{\partial \lambda_i} = 0 \tag{3.4.2}$$

$$\sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s}) = \tilde{p}(g_i) \tag{3.4.3}$$

Where equation (3.4.3) means that the expectation of the characterizing functions of the constructed MaxEnt model must equal the expectation of the characterizing functions over the empirical distribution $\tilde{p}$. This result reflects the imposed constraints in the dual MaxEnt problem.

### 3.4.2 Non-Linear Numerical Optimization

Probabilistic ANNs are a particular realization of undirected graphical models (i.e. MaxEnt models or Markov random fields (MRF) )[Lauritzen, 1996]. Over the last two decades, there has been extensive research into the training of ANNs using *Numerical Optimization* methods. GRadient Ascent (GRA) and its variants, Newton's method, limited memory quasi-Newton's method (L-BFGS), and Conjugate Gradient (CG) methods are common algorithms to train MaxEnt/CRF models (see [Haykin, 1998, Bishop, 1995, Nocedal and Wright, 1999]). The HMMs can be trained as undirected graph models using numerical methods [Kapadia, 1998]. This can be achieved by relaxing the probabilistic constraints during the training process (i.e. unconstrained parameters). A softmax parameter transformation is used to maintain the probabilistic constraints while updating the parameters. However, maximizing the MMIE/CML objective function using the EBW algorithm is faster than gradient-based methods for HMM models [Gopalakrishnan et al., 1991, Normandin, 1991, Kapadia et al., 1993].

These methods rely on local quadratic approximation by expanding the CML *nonlinear* objective function $\mathcal{F}_{\mathrm{CML}}(\Lambda+\delta)$ using Taylor expansion around the current model point $\Lambda$ in parameter space and given by

$$\mathcal{F}_{\mathrm{CML}}(\Lambda+\delta) \approx L(\Lambda) + \delta^T \mathbf{g}(\Lambda) + \frac{1}{2}\delta^T \mathbf{H}(\Lambda)\delta + \dots \tag{3.4.4}$$

where $\mathbf{g}(\Lambda)$ is the local gradient vector defined by

$$\mathbf{g}(\Lambda) = \frac{\partial \mathcal{F}_{\text{CML}}(\Lambda)}{\partial \lambda_i}\Big|_{\Lambda} \tag{3.4.5}$$

and the $\mathbf{H}(\Lambda)$ is the local Hessian matrix defined by

$$\mathbf{H}_{ij}(\Lambda) \equiv \frac{\partial \mathcal{F}_{\text{CML}}(\Lambda)}{\partial \lambda_i \partial \lambda_j}\Big|_{\Lambda} \tag{3.4.6}$$

By ignoring the second order derivative, a first order approximation of the CML will lead to the GRA methods and their variants as described in Section 3.4.2. Newton's method and its variants use second order derivative quadratics for fast rate of convergence with the additional cost of calculating the Hessian information as described in Section 3.4.2.

The performance of the numerical optimization methods is task dependent. The usefulness of using Newton, L-BFGS, CG methods over simple on-line GRA algorithms may be limited to train complex architectures of ANNs [LeCun et al., 1998a]. However, for high dimensional binary spaces used in natural language processing, methods like L-BFGS or CG show clear success but the models were trained with many iterations ($> 50$) [Malouf, 2002, Sha and Pereira, 2003].

### Linear Approximation

The gradient of the MaxEnt models is given by

$$\mathcal{F}_{\text{CML}}(\Lambda) = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}, \mathbf{s}) \ln P_{\Lambda}(\mathbf{s} \mid \mathbf{o}) \tag{3.4.7}$$

$$\mathbf{g}_i(\Lambda) = \frac{\partial \mathcal{F}_{\text{CML}}(\Lambda)}{\partial \lambda_i} = \tilde{p}(g_i) - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_{\Lambda}(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o}, \mathbf{s}) \tag{3.4.8}$$

and the GRA update is given by

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} + \eta \mathbf{g}(\Lambda) \tag{3.4.9}$$

The step size $\eta$ must be small enough to ensure a stable increase of the CML objective function. It can be shown that the algorithm is convergent provided that $\eta$ satisfies the condition $0 < \eta < \frac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of the Hessian matrix $\mathbf{H}(\mathbf{\Lambda}^*)$ evaluated at the global maximum of the CML objective function [Haykin, 1998]. In practice, second order statistics are not accumulated so $\lambda_{\max}$ is not known and $\eta$ is chosen in ad-hoc fashion by trial and error.

For large redundant databases, *on-line* GRA methods show increased rates of convergence for ANN training [Robinson, 1994, LeCun et al., 1998a]. In such methods, the training data is divided into blocks and the update is repeated after each block to increase the rate of convergence.

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} + \eta \mathbf{g}^{\text{blk}}(\Lambda) \tag{3.4.10}$$

On-line methods are based on the basic fact that the objective function is actually summation of objective functions associated with every observation $\mathbf{o}$. The theoretical justification of on-line or *sequential* estimation is related to the Robbins-Monro algorithm [Bishop, 1995]. The effectiveness of on-line methods depends on some factors related to the model, the actual data redundancy, and the objective function. Hence, it is task dependent and the actual gain from these methods with respect to batch training must be evaluated experimentally. It is important to note that online methods are general approaches and are not related only to the *batch* first order approximation.

Many authors have described variants of the GRA algorithm, where increased rates of convergence through learning rate adaptation was proposed [Jacobs, 1988, Sutton, 1992, Murata et al., 1997]. We refer to this family of algorithms as Adaptive GRA or GRA-adaptive. Adaptive GRA and its variants including the on-line versions were used to train the most successful Hybrid ANN/HMM speech recognition systems including ABBOT [Kershaw et al., 1996, Robinson, 1994]. Our implementation is similar to a particular implementation of the GRA algorithm [Robinson, 1994], where the updates are given by

$$\eta_i^{(\tau)} = \begin{cases} \min(\eta_i^{(\tau-1)}\phi, \eta_{\max}) & \mathbf{g}^{(\tau)}(\Lambda)\bar{\mathbf{g}}^{(\tau-1)}(\Lambda) > 0 \\ \max(\eta_i^{(\tau-1)}\kappa, \eta_{\min}) & \mathbf{g}^{(\tau)}(\Lambda)\bar{\mathbf{g}}^{(\tau-1)}(\Lambda) < 0 \end{cases} \tag{3.4.11}$$

where

$$\bar{\mathbf{g}}^{(\tau)}(\Lambda) = (1-\theta)\mathbf{g}^{(\tau)}(\Lambda) + \theta\bar{\mathbf{g}}^{(\tau-1)}(\Lambda) \tag{3.4.12}$$

and

$$\lambda_i^{(\tau)} = \begin{cases} \lambda_i^{(\tau-1)} + \eta_i^{(\tau)} & \mathbf{g}^{(\tau)}(\Lambda) < 0 \\ \lambda_i^{(\tau-1)} - \eta_i^{(\tau)} & \mathbf{g}^{(\tau)}(\Lambda) > 0 \end{cases} \tag{3.4.13}$$

The learning parameters were set as follows $\kappa = 0.9$, $\phi = \frac{1}{\kappa}$, $\theta = 0.5$, and $\eta_i^{(0)} = \eta$. The Resilient Propagation (RProp) algorithm [Riedmiller and Braun, 1993], which also uses a Manhattan update rule, is conceptually similar to the described algorithm but it does not involve the gradient average step in equation (3.4.12). The Manhattan update rule does not involve the gradient magnitude.

### Quadratic Approximation

Newton's method can be used to estimate MaxEnt models based on local quadratic approximation of the CML objective function. The local Hessian matrix $\mathbf{H}(\Lambda)$ of the MaxEnt models is defined by

$$\begin{aligned} \mathbf{H}_{ij}(\Lambda) &= -\Big(E_\Lambda(g_i(\mathbf{o},\mathbf{s})g_j(\mathbf{o},\mathbf{s})) - E_\Lambda(g_i(\mathbf{o},\mathbf{s}))E_\Lambda(g_j(\mathbf{o},\mathbf{s}))\Big) \\ &= -\operatorname{Cov}\Big(g_i(\mathbf{o},\mathbf{s}), g_j(\mathbf{o},\mathbf{s})\Big) \end{aligned} \tag{3.4.14}$$

and

$$\mathbf{H}(\mathbf{\Lambda}) = -\begin{pmatrix} \mathrm{var}_{11} & \mathrm{cov}_{12} & \dots \\ \mathrm{cov}_{21} & \mathrm{var}_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Since a MaxEnt model is a solution for a concave function, the Hessian must be negative definite.[2] The Newton's Method update rule is given by

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} - \eta^{(\tau)}\mathbf{H}^{-1}(\Lambda)\mathbf{g}(\Lambda) \tag{3.4.15}$$

Since CML is not a quadratic function, taking the full Newton step $\mathbf{H}^{-1}(\Lambda)\mathbf{g}(\Lambda)$ may lead to an overshoot of the maximum. Hence, $\eta^{(\tau)} \neq 1$ will lead to the *damped* Newton step. A *line search* algorithm is used to calculate $\eta^{(\tau)}$. A line search works by evaluating the objective function starting from the current model in the direction of search and choosing $\eta^{(\tau)}$ will lead to an increase of the CML objective function. Line search algorithms can also be applied to the GRA algorithms.

Hessian matrix calculation, its inverting and storage, makes Newton's Method useful only for small scale problems. Quasi-Newton or variable metric methods can be used when it is impractical to evaluate the Hessian matrix. Instead of obtaining an estimate of the Hessian matrix at a single point, these methods gradually build up an approximate Hessian matrix by using gradient information from some or all of the previous iterates visited by the algorithm. Limited memory quasi-Newton's methods like L-BFGS are particular realizations of quasi-Newton's methods that cut down the storage for large problems [Nocedal and Wright, 1999]. L-BFGS optimization methods are the state-of-the-art training method for MaxEnt optimization for natural language processing [Malouf, 2002, Sha and Pereira, 2003].

[2] The training of MaxEnt models usually leads to intrinsic ill-conditioned training problems and there is no guarantee that the Hessian matrix will be negative definite.

Quick Propagation (QProp) is also a second order approximation for large scale problems, where the exact Hessian matrix is replaced with a finite difference diagonal approximation [Fahlman, 1988]. The update rule is given by

$$\lambda^{(\tau+1)} = \lambda^{(\tau)} - \eta^{(\tau)} \Big( \frac{\mathbf{g}^{(\tau)}(\Lambda) - \mathbf{g}^{(\tau-1)}(\Lambda)}{\lambda^{(\tau)} - \lambda^{(\tau-1)}} \Big)^{-1} \mathbf{g}^{(\tau)}(\Lambda) \tag{3.4.16}$$

A fast *on-line* QProp algorithm (the Manhattan QProp), where its update rule does not involve the gradient magnitude, to update the HMMs parameters was developed by [Kapadia, 1998].

The CG method belongs to second order optimization and it uses conjugate directions to the search directions from the previous iteration to accelerate the slow convergence rates associated with GRA algorithms while avoiding the computational cost and storage required for Newton's methods [Shewchuk, 1994].

### 3.4.3 Lower Bound Optimization

This thesis relies on training MaxEnt/CRF models using lower bound optimization since these methods usually tend to be less heuristic than numerical optimization methods. Exact lower bound optimization is sometimes slow and in our work, we derive a family of Iterative Scaling (IS) algorithms, which we call *Approximate Iterative Scaling* (AIS) to speed up the training process. The key idea of the family of AIS algorithms is to understand how the models and observation spaces are constructed and try to integrate this prior knowledge into the training algorithms, without departing totally from principled optimization, which is the main idea behind lower bound optimization. The AIS algorithms increase the rates of convergence by assigning different learning rates for each state or for individual Lagrange multipliers using the learning rate adaptation algorithm (see algorithm 3.2).

The classical estimation method for the Lagrange's multipliers, which are the parameters of the MaxEnt distribution, is the Generalized Iterative Scaling (GIS) algorithm developed by Darroch and Ratcliff [Darroch and Ratcliff, 1972]. The convergence proof of the GIS algorithm is conditioned on the assumption that the summation of the calculated constraints for each observation must be constant ( $\sum_i g_i(\mathbf{o}) = C_{\text{gis}}$ and $g_i(\mathbf{o}) > 0$ ) for each observation. Hence, a correction feature is added to satisfy that condition [Ratnaparkhi, 1997]. The learning speed of the GIS is inversely proportional to the value of the scaling factor used to calculate the correction feature.

More recently, the Improved Iterative Scaling (IIS) algorithm has been developed [Della Pietra et al., 1997, Berger, 1997]. The basic idea behind the IIS algorithm is to make use of an auxiliary function, which bounds the change in CML from below after each iteration. The use of *auxiliary function* or *lower bound* is the standard means of justifying and implementing the EM algorithm [Dempster et al., 1977]. The IIS algorithm is a direct application of lower bound optimization for exponential models. It can be shown that the IIS algorithm is an improvement on the GIS algorithm, in the sense that it does not restrict the convergence on the condition $\sum_i g_i(\mathbf{o}) = C_{\text{gis}}$ and hence removes the need for a correction feature during the training process. The condition $g_i(\mathbf{o}) > 0$ is necessary for the IIS algorithm proof. The convergence rate of the IIS algorithm may be faster than the GIS algorithm for some tasks but since the IIS algorithm relaxes the condition $\sum_i g_i(\mathbf{o}) = C_{\text{gis}}$, the IIS algorithm requires a numerical search to calculate the step for each iteration using Newton's method as an additional overhead.

First order statistics usually imply negative values and the condition $g_i(\mathbf{o}) > 0$ cannot be valid. To overcome this problem, we developed the *Generalized Improved*

*Iterative Scaling (GIIS) algorithm* to support $g_i(\mathbf{o}) < 0$. The GIIS algorithm[3] inherits most of the IIS algorithm derivation and supports $g_i(\mathbf{o}) < 0$ via a mathematical trick published in [Collins et al., 2000]. Its derivation for random fields can be found in the technical note [Hifny, 2002] and the conditional version of the algorithm is detailed in the following subsection.

The GIS, IIS, and GIIS algorithms are a family of algorithms that prove the convergence of the training iteratively. We refer to this family of algorithms as *exact* Iterative Scaling (IS) algorithms. On the other hand, we derive a family of algorithms, which do not guarantee the convergence of the training process but they usually provide fast and stable training. This family of algorithms is referred to as Approximate Iterative Scaling (AIS) algorithms (see subsection "Approximate Iterative Scaling" on page 60).

### Iterative Scaling Optimization

The purpose of the scaling algorithms GIIS and IIS is to estimate the parameters $\Lambda$ using an auxiliary function approach. IIS uses Jensen's inequality to decouple all parameters in order to solve a single Newton equation for each individual parameter independently. It establishes a lower bound on the change of the likelihood as follows:

$$\begin{aligned}\Delta\mathcal{L} &= \mathcal{L}(\tilde{p} \mid p_{\lambda+\delta}) - \mathcal{L}(\tilde{p} \mid P_\Lambda)\\ &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \ln p_{\lambda+\delta}(\mathbf{s} \mid \mathbf{o}) - \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \ln P_\Lambda(\mathbf{s} \mid \mathbf{o})\\ \Delta\mathcal{L} &\geq \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \delta_i g_i(\mathbf{o},\mathbf{s}) - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log \frac{Z_{\lambda+\delta}(\mathbf{o})}{Z_\lambda(\mathbf{o})}\end{aligned}$$

[3] The main idea behind the GIIS algorithm was suggested by Lafferty [Lafferty, 2002].

by applying the inequality $-\ln\alpha \geq 1-\alpha$ for all $\alpha > 1$, a lower bound on the change of the likelihood is established. Hence,

$$\begin{aligned}
\Delta\mathcal{L} &\geq \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \delta_i g_i(\mathbf{o},\mathbf{s}) + 1 - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \frac{Z_{\lambda+\delta}(\mathbf{o})}{Z_\lambda(\mathbf{o})} \\
&\geq \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \delta_i g_i(\mathbf{o},\mathbf{s}) + 1 - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \frac{\sum_{\mathbf{s}} \exp\Big(\sum_i (\lambda_i+\delta_i) g_i(\mathbf{o},\mathbf{s})\Big)}{\sum_{\mathbf{s}} \exp\Big(\sum_i \lambda_i g_i(\mathbf{o},\mathbf{s})\Big)} \qquad (3.4.17) \\
&\geq \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \delta_i g_i(\mathbf{o},\mathbf{s}) + 1 - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \exp \sum_i \delta_i g_i(\mathbf{o},\mathbf{s})
\end{aligned}$$

let

$$g_i(\mathbf{o}) = \text{sign}(g_i(\mathbf{o}))|g_i(\mathbf{o})| = s_i(\mathbf{o})|g_i(\mathbf{o})| \qquad (3.4.18)$$

and

$$M(\mathbf{o},\mathbf{s}) = M(\mathbf{o}) = \sum_i |g_i(\mathbf{o})| \qquad (3.4.19)$$

hence,

$$\begin{aligned}
\frac{Z_{\lambda+\delta}(\mathbf{o})}{Z_\lambda(\mathbf{o})} &= \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \exp \frac{\sum_i \delta_i s_i(\mathbf{o})|g_i(\mathbf{o},\mathbf{s})| M(\mathbf{o})}{M(\mathbf{o})} \\
&\leq \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \sum_i \frac{|g_i(\mathbf{o},\mathbf{s})|}{M(\mathbf{o})} \exp \delta_i s_i(\mathbf{o}) M(\mathbf{o})
\end{aligned} \qquad (3.4.20)$$

where $|g_i(\mathbf{o},\mathbf{s})|/M(\mathbf{o})$ is a p.m.f over $i$, $\sum_i |g_i(\mathbf{o},\mathbf{s})|/M(\mathbf{o}) = 1$ and equation (3.4.20) is a direct application of Jensen's inequality

$$\exp\{E(q(\mathbf{z}))\} = \exp \sum_{\mathbf{z}} p(\mathbf{z})q(\mathbf{z}) \leq \sum_{\mathbf{z}} p(\mathbf{z}) \exp q(\mathbf{z})$$

From equations (3.4.17, 3.4.20):

$$\begin{aligned}
\Delta\mathcal{L} \geq& \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_i \delta_i g_i(\mathbf{o},\mathbf{s}) + 1 \\
&- \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \sum_i \frac{|g_i(\mathbf{o},\mathbf{s})|}{M(\mathbf{o})} \exp \delta_i s_i(\mathbf{o}) M(\mathbf{o}) \qquad (3.4.21)
\end{aligned}$$

$$\begin{aligned}\frac{\partial \Delta\mathcal{L}}{\partial \delta_i} &= 0 \\ &= \tilde{p}(g_i) - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \frac{|g_i(\mathbf{o},\mathbf{s})|}{M(\mathbf{o})} s_i(\mathbf{o}) M(\mathbf{o}) \exp \delta_i s_i(\mathbf{o}) M(\mathbf{o})\end{aligned} \tag{3.4.22}$$

Hence,

$$\sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s}) \exp \delta_i s_i(\mathbf{o}) M(\mathbf{o}) = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) g_i(\mathbf{o},\mathbf{s}) = \tilde{p}(g_i) \tag{3.4.23}$$

Equation (3.4.23) leads to the iterative algorithm 3.1. Clearly, if every $s_i(\mathbf{o})$ is positive, the GIIS algorithm, will be exactly the same as the IIS algorithm. We can prove that equation (3.4.23) has a unique solution by calculating the second derivative and checking its convexity.

**Algorithm 3.1** The Generalized Improved Iterative Scaling (GIIS) algorithm

```
initialize $\lambda_i = 0.0$, $i = 1, 2, \ldots, n$. {uniform model over states}
initialize $\mathcal{I}$ with the maximum iteration count.
let $M(\mathbf{o}) = \sum_i |g_i(\mathbf{o})|$, $s_i(\mathbf{o})$ is the sign of $g_i(\mathbf{o})$.
repeat
    $\tau = \tau + 1$
    $P_\Lambda(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_\Lambda(\mathbf{o})} \exp\left(\sum_i \lambda_i g_i(\mathbf{o},\mathbf{s})\right)$
    Solve $\sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}) P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s}) \exp\left(\delta_i^{(\tau)} s_i(\mathbf{o}) M(\mathbf{o})\right) = \tilde{p}(g_i)$
    $\lambda_i^{(\tau)} = \lambda_i^{(\tau-1)} + \delta_i^{(\tau)}$
until $\mathcal{F}_{\mathrm{CML}}(\Lambda) < \Delta$ or $\tau < \mathcal{I}$
```

### Efficient Iterative Scaling

In general, iterative scaling or lower bound optimizers decouple all model parameters to find efficient update equations. Hence, they are similar to the linear approximation of gradient based optimizers. As a result, the input features should be decorrelated for faster convergence. The PCA algorithm can be used to remove the linear correlation in the input features.[4] On the other hand, shifting the input features to have zero means, and scaling them to have equal variances usually enhances the convergence properties of the training algorithms. These ideas as they relate to gradient based optimization for ANNs are detailed in [LeCun et al., 1998b].

Based on our empirical observations, the GIIS algorithm is not efficient for speech recognition problems for the following reasons:

1. GIIS can lead to faster steps only if the absolute feature sum, $M(\mathbf{o})$, varies dramatically over the observations. This means that $M(\mathbf{o})$ as a random variable has a large variance. Unfortunately, this is not the case for the spaces based on continuous observations developed in this thesis.

2. An iteration of the GIIS algorithm has an inner loop over the training corpus to calculate GIIS steps using Newton's method. Hence, it is computationally expensive and it is impractical for speech recognition applications.

3. It is difficult to implement it for high performance parallel environments.

However, the derivation of the GIIS algorithm will allow us to derive simple formulae to train MaxEnt models such as GIS training [Darroch and Ratcliff, 1972] without adding the correction feature commonly used to relax the constraint that summation of the features needs to be constant [Ratnaparkhi, 1997]. Let $C = \max_o M(\mathbf{o})$ be the maximum value of the absolute feature summation of each observation in the training corpus[5], the GIIS can be simplified to avoid root finding and provide two formulae to estimate the models.

[4] The last stage of the MFCC front end processing is a DCT projection, which approximates a PCA rotation [Fukunaga, 1990]. As a result, the MFCC features may be suitable for model training based on lower bound optimizers such as the IS algorithm.

[5] The reader should note that $C \neq C_{\text{gis}}$.

The general formula is useful because the first order statistics usually have some negative values. In order to derive an efficient estimation equation let $x_i = \exp\left(\delta_i^{(\tau)} C\right)$, if $a = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}) P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s})$ is the expectation of the observations that have positive values, $c = \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}) P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s})$ is the expectation of the observations that have negative values, and $b = \tilde{p}(g_i)$, then we can rewrite the GIIS equation as

$$\begin{aligned} \tilde{p}(g_i) &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o}) P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o},\mathbf{s}) \exp\left(\delta_i^{(\tau)} s_i(\mathbf{o}) M(\mathbf{o})\right) \\ 0 &= a x_i + c\frac{1}{x_i} - b \\ 0 &= a x_i^2 - b x_i + c \end{aligned} \tag{3.4.24}$$

By solving the quadratic equation (3.4.24), $\delta_i = \log(x_i)$ is obtained, where $x_i > 0$; hence, $\lambda_i^{(\tau)} = \lambda_i^{(\tau-1)} + \delta_i^{(\tau)}$. This formula does not arise in the MaxEnt estimation problems. It is important to note that constraints, which have negative values need to accumulate two different expectations $a$ and $c$. So, the storage complexity is doubled for any feature that has negative values. For this reason, it is usually efficient to avoid formulating problems based on constraints which have negative values. Hence, equation (3.4.24) is the general form used to update the parameters and it is suitable for any constraints. Equation (3.4.25) is a special case from equation (3.4.24) and it is suitable only for constraints that have positive values.

$$\lambda_i^{(\tau)} = \lambda_i^{(\tau-1)} + \frac{1}{C} \log \frac{\tilde{p}(g_i|\mathbf{o})}{P_\Lambda(g_i|\mathbf{o})} \tag{3.4.25}$$

Despite the fact that these formulae do not take the full GIIS step towards the global solution, there is no practical difference in the training speed in our work between using these methods and the actual GIIS algorithm and they are our choice in this thesis. Note that, for *normalized spaces*, where for *every* observation in the training corpus, $M(\mathbf{o}) = 1$ or $M(\mathbf{o}) = C$, the speed of the GIIS and GIS are the same and these formulae are ideal for updating the parameters. The normalized spaces are addressed in detail in Chapter 6.

If we can only update a single Lagrange multiplier at a time (sequential or axis update), then $M(\mathbf{o}) = 1$ if the maximum value of this constraint in the training corpus $\leq 1$ and faster IS steps can be achieved by updating one coordinate at a time. However, this approach is useful only for very small problems and we did not observe any significant improvement in the results obtained with this approach. Of course, this approach is impossible for speech applications so updating all the Lagrange multiplier (parallel update) at the same time has been chosen as it is the most effective approach for speech recognition problems.

### On-line Iterative Scaling

During an early stage of this thesis, we investigated the use of on-line or incremental estimation of MaxEnt Models. Blocks of the training samples were selected and represented sequentially to update the models via the IS algorithm.

It is clearly shown in Figure 3.1 that on-line estimation can lead to a faster training procedure, where two iterations were sufficient to increase CML to estimate quadratic classifier for the TIMIT database.

As mentioned before, on-line methods are a general approach and can be associated with any learning algorithm. However, on-line methods do not encourage addressing fast batch algorithms. On the other hand, it is common to take advantage of parallel computing, to distribute the calculations of the expectations over a parallel environment. Hence, the batch update is the ideal method for training large databases in speech recognition even we use on-line methods over parallel environments since

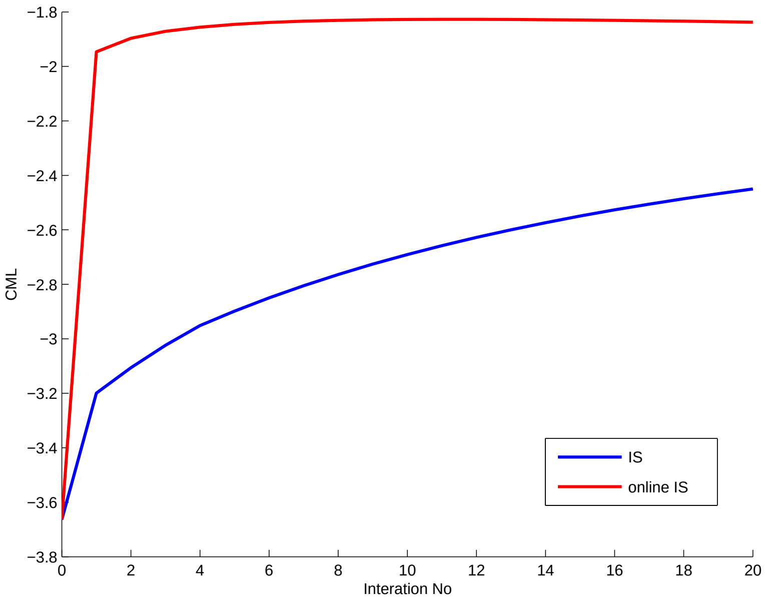


Figure 3.1: Fast estimation of a quadratic MaxEnt classifier for TIMIT task using on-line methods.

on-line methods will raise issues related to the ideal block size in terms of utterances, overfitting training data and complexity of the update procedures [Robinson, 1994].

In fact, these methods may be very useful for certain type of problems - the speaker adaptation problem for example - where we would like to adapt models for certain speakers or environments where there are usually well trained models and few adaptation data.

On the other hand, there was a major contribution of that experimental work. It sheds the light on a basic fact, that we can achieve faster training algorithms by changing the step size from the theoretical value, $C = \max_o M(\mathbf{o})$, that guarantees convergence of the CML objective function. This observation will lead to faster training methods as detailed in following section. It can be easily shown that taking different steps for each state or Lagrange multiplier can lead to automatically adapted steps to provide faster Approximate Iterative Scaling.

### Approximate Iterative Scaling

The training speed of the IS algorithms is measured by the slowing factor $C = \max_o M(\mathbf{o})$. As shown above, $C$ will guarantee the convergence of the training process but usually at a price of slow training speed. As was shown in the previous section, it is easy to show experimentally that faster training can be achieved if we slow the IS algorithm by a factor $D_{\text{AIS}}$[6] only, where $0 < D_{\text{AIS}} \ll C$. If the value of $D_{\text{AIS}}$ is sufficiently small, we expect a faster IS algorithm where the value of CML will increase at every iteration but there is no theoretical guarantee that the algorithm will converge. AIS algorithms can help to set different state or parameter level learning rates. The update rule is given by

$$\lambda_i^{(\tau)} = \lambda_i^{(\tau-1)} + \frac{1}{D_{\text{AIS}}} \log \frac{\tilde{p}(g_i|\mathbf{o})}{P_\Lambda(g_i|\mathbf{o})} \tag{3.4.26}$$

Most researchers train the MaxEnt models by numerical optimization methods as the exact IS algorithm seems to be inefficient for their tasks. In this thesis, all models were trained by these AIS algorithms which provided fast training without relying too much on the numerical optimization techniques.

The question is how to select $D_{\text{AIS}}$ value to avoid the heuristics commonly known in GRA methods to find a good initial value for $\eta$. Some ideas that can lead to a general AIS algorithm at the parameter level will be detailed. In Chapters 5 and 6, we will detail some techniques that were crucial to training sequential CRF models.

The first observation is related to the basic fact, that some constraints do not contribute to the calculation of the state scores. This essentially means that we

[6] In the thesis, the letter $D$ , will refer to the AIS constant and letter $C$ will refer to the exact IS constant.

can have a state dependent scaling factor. The value of this scaling factor must be calculated for each state and all parameters belonging to this state are updated using this constant. For some tasks, it is easy to show that this value is not approximate but exact and prove the convergence of the learning process. However, to keep the concept general, we select a learning factor for each state that we consider an approximate value. $D_{\mathrm{AIS}}$ is given by

$$D^{(\mathbf{s})}_{\max} = \max_{g \in s} M(\mathbf{o}) \tag{3.4.27}$$

The second takes advantage of the fact, that $D^{(\mathbf{s})}_{\max}$ is too conservative if the variance of $M(\mathbf{o})$ is high. So, the scaling factor may reflect rare samples in the training corpus. Note that, the IIS algorithm may be developed to handle large variance problems [Della Pietra et al., 1997]. So by smoothing all the scaling constants of the observations of each state, we may have a more objective measure of $D_{\mathrm{AIS}}$. The update rule is given by

$$D^{(\mathbf{s})}_{\mathrm{mean}} = \mathrm{E}_{g \in s}(M(\mathbf{o})) \tag{3.4.28}$$

$D^{(\mathbf{s})}_{\mathrm{mean}}$, and $D^{(\mathbf{s})}_{\max}$ work at the state level and usually provide faster training without any additional cost. If the states of the model are associated with all constraints, we can try to set each state to the global mean step (i.e. $D^{(\mathbf{s})}_{\mathrm{mean}} = D^{(\mathcal{D})}_{\mathrm{mean}} = \mathrm{E}_{g \in \mathcal{D}}(M(\mathbf{o}))$). These results pointed to the possibility of updating each parameter independently. This will lead to the adaptive AIS training.

To work at parameter level, we need to check the sign of the gradients of a parameter on two consecutive iterations. If the two gradients have similar signs (i.e. $\mathbf{g}^{(\tau)}(\Lambda)\mathbf{g}^{(\tau-1)}(\Lambda) > 0$), the $D$ constant of that parameter will be decreased to accelerate convergence in the regions with low curvature. When the sign of the derivative of a parameter oscillates (i.e. $\mathbf{g}^{(\tau)}(\Lambda)\mathbf{g}^{(\tau-1)}(\Lambda) < 0$)), the parameter is in the vicinity of the maximum point and the parameter missed the optimal point due to the high curvature regions. Therefore, that parameter $D$ must increased or be assigned to the more conservative value associated with its state $D^{(\mathbf{s})}_{\text{mean}}$. This idea will lead to version of an adaptive AIS that updates the Lagrange multipliers with individual learning rates.

**Algorithm 3.2** Fast adaptive AIS algorithm with individual learning rate

```
initialize $\mathcal{I}$ with the maximum number of iterations.
initialize $D^{(0)}_{ji} = D^{(\mathbf{s}=j)}_{\text{mean}}$ or $D^{(\mathbf{s}=j)}_{\text{max}}$, $0.8 < \kappa < 0.9$.
repeat
    $$\mathbf{g}(\Lambda) = \tilde{p}(g_i) - \sum_{\mathbf{o}} \tilde{p}(x) \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{o}) g_i(\mathbf{o}, \mathbf{s}) \qquad \forall i$$
    $$D^{(\tau)}_{ji} = \begin{cases} \max(D^{(\tau-1)}_{ji}\kappa, 1.0) & \mathbf{g}^{(\tau)}(\Lambda)\mathbf{g}^{(\tau-1)}(\Lambda) > 0 \\ D^{(\mathbf{s}=j)}_{\text{mean}} & \mathbf{g}^{(\tau)}(\Lambda)\mathbf{g}^{(\tau-1)}(\Lambda) < 0 \end{cases}$$
    $$\lambda^{(\tau)}_{ji} = \lambda^{(\tau-1)}_{ji} + \frac{1}{D^{(\tau)}_{ji}} \log \frac{\tilde{p}(g_{ji}|\mathbf{o})}{P_\Lambda(g_{ji}|\mathbf{o})}$$
until $\mathcal{F}_{\text{CML}}(\Lambda) < \Delta$ or $\tau < \mathcal{I}$
```

Algorithm 3.2 summarizes the idea described. Where it is clear that the algorithm tries to change $(D^{(\mathbf{s})}_{\text{mean}} < D^{(\lambda)}_{\text{AIS}} < 1.0)$, $D^{(\lambda)}_{\text{AIS}} = D^{(\mathbf{s})}_{\text{mean}}$ is the most principled value to set the slowing constant, $D$ of each parameter. A value $D^{(\lambda)}_{\text{AIS}} = 1$ reflects an ideal step taken for that parameter without any slowing. The algorithm has only one heuristic parameter $\kappa$ that is usually set to 0.9 and works with minor additional storage cost of saving the gradient at each iteration. When the sign of the derivative of a parameter oscillates (i.e. $\mathbf{g}^{(\tau)}(\Lambda)\mathbf{g}^{(\tau-1)}(\Lambda) < 0$)), the algorithm selects the conservative value associated with its state $D^{(\mathbf{s})}_{\text{mean}}$.[7]

[7] It can be set to $\min(D^{(\tau-1)}_{i}\phi, D^{(\mathbf{s}=j)}_{\text{mean}})$ and $1.1 < \phi < 2$.

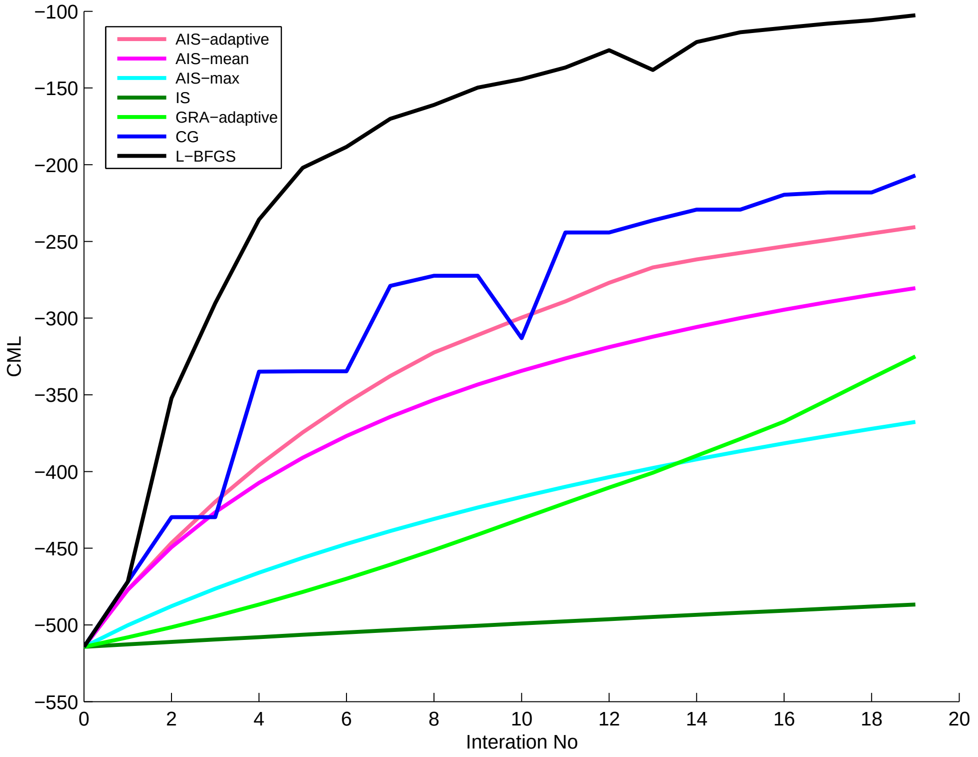


Figure 3.2: Deterding task optimization.

### 3.4.4 Optimization Evaluation

As an example of the AIS algorithm's speed described in this thesis, we train a conditional random field kernel classifier on Deterding's vowel classification task [Deterding, 1989]. The classifier implementation and the task is detailed in Chapter 4. Figure 3.2 and its companion Table 3.5 show the training speed for several algorithms for the CML objective function and the associated Classification Accuracy (CA).

Figure 3.2 shows that the L-BFGS very quickly increases the CML objective function. Moreover, the L-BFGS and CG algorithms do not guarantee the increase of the CML objective function at every iteration. This is due to the fact that these

Table 3.5: *Classification accuracy for Deterding's vowel task. The training was stopped after 20 iterations.*

| algorithm | CA |
|---|---|
| IS | 54.3% |
| CG | 55.8% |
| L-BFGS | 56.4% |
| AIS-max | 56.4% |
| GRA-adaptive | 59.9% |
| AIS-mean | 60.1% |
| AIS-adaptive | 62.9% |

methods approximate the CML locally with a quadratic approximation, which may fail at some iterations because linear and quadratic approximations do not provide a lower bound for the objective function. However, one cannot draw conclusions from the evaluation of the training algorithms. Unfortunately, there is no direct relation between the CML objective function and the CA criterion, even though maximizing the posterior probability provides a lower bound for the CA criterion. Table 3.5 shows that the adaptive AIS algorithm leads to significant accuracy improvements with respect to all algorithms.

For the gradient family optimization, the adaptive GRA algorithm is able to outperform the L-BFGS and CG algorithms. This may explain why adaptive GRA algorithms are popular in ANNs based speech recognition community. We did not report results for the QProp algorithm.

For the IS family of algorithms, it is clear that the exact IS algorithm is the slowest algorithm in terms of increasing of the CML objective function and its CA is the worst one. By exploring the details of the observation space, it was easy to have faster algorithms. AIS-max is the slowest in terms of the training speed, AIS-mean is faster and AIS-adaptive is the fastest AIS algorithm since it is an adaptive version of AIS-mean. Indeed, this gradual increase in the training speed is reflected in the CA results, where the AIS-adaptive algorithms lead to the best CA reported in this work. These results show that blind optimization by exact IS algorithms is not the right way to use these algorithms. Prior knowledge about the tasks and how their observation spaces were constructed will usually lead to fast algorithms. To compare the two optimization families, adaptive GRA may be considered an effective training algorithm for small tasks but it is highly dependent on the initial $\eta$ as described in Section 3.4.2. The AIS-adaptive and AIS-mean algorithms outperformed the adaptive GRA and always have a clear way to specify their $D_{\mathrm{AIS}}$. The AIS-max and AIS-mean set the $D_{\mathrm{AIS}}$ at the state level and AIS-adaptive sets $D_{\mathrm{AIS}}$ at the parameter level.

Of course, there is a question of why the lower bound optimization methods were able to outperform gradient based methods on this task. The lower bound methods provide consistent approximate optimization for CML objective function regardless of the complexity of the task and models, and perhaps this may explain their good performance here. On the other hand, numerical optimization methods depend on local quadratic approximation around the current solution, which may be poor for approximating the nonlinear CML objective function for this task. On the other hand, the relationship between the input features may affect the training speed of any algorithm.

In fact, our results based on lower bound optimization are not reported for the high dimensional binary observation spaces, which have highly correlated features.[8] These results are well established in the speech research community, where the EM

[8] We argue that with the family of AIS algorithms, the training speed may be much better than the reported results but we did not address these problems in this thesis.

and EBW algorithms, which are lower bound optimizers, are well known to be the fastest training algorithms with respect to any gradient based optimization. On the other hand, CML optimization for e-family distributions is based on the conditional expectation maximization (CEM) algorithm, which is mainly derived based on bounded optimization [Jebara, 2002]. Of course, the reported results cannot be generalized, however it explains why AIS is the main optimization tool in this thesis. Note that, numerical optimization based on Taylor's approximations are more general methods than bound optimization, which take advantage of the structure of the objective function as a special case.

## 3.5 Maximum Entropy and Regularization

Generally, parameter estimation for MaxEnt models from finite training data is an ill-posed inverse problem since there is an infinite number of solutions if only constraints from data are used. Additional prior information or assumptions are needed to *smooth* and guarantee the uniqueness of the solution and make the problem well-posed [Haykin, 1998, page 266].

*Regularization* of an ill-posed problem is common solution to overcome poor generalization problem and provide effective complexity control. Regularization is achieved by adding a penalty term to the CML criterion, resulting in the following cost function:

$$\Lambda^* = \text{argmax}_\Lambda \mathcal{F}_{\text{CML}}(\Lambda) - \sum_q \alpha_q \Omega_q(\Lambda), \tag{3.5.1}$$

where adding the penalty term $\sum_q \alpha_q \Omega_q(\Lambda)$ will have large values for complex systems. As a result, it prevents the estimated parameters to have large values and prevents the solution to be tuned to the random noise on the training data. State-of-the-art regularizers may measure the complexity of a model with a *mixture* of different regularizers given by

$$\sum_q \alpha_q \Omega_q(\Lambda) = \sum_q \alpha_q \sum_\Lambda \|\lambda\|^q \tag{3.5.2}$$

where for different $q \in \{1, 2\}$, we get different kinds of Minkowski norm for the Lagrange multipliers values. The $\Omega_q(\Lambda)$ is usually explained as imposing a prior distribution over the model parameters in the Bayesian framework.

The $l_2$ regularizer, where $\Omega_2(\Lambda) = \sum_\Lambda \|\lambda\|^2$, is the classic choice and it implies zero mean Gaussian priors over the model parameters in the Bayesian setting [Bishop, 1995, Chen and Rosenfeld, 1999]. However, the Gaussian prior does not lead to a sparse solution as the parameter values do not approach zero after the training procedure.

The $l_1$ regularizer or Lasso penalty, where $\Omega_1(\Lambda) = \sum_\Lambda ||\lambda||$ is often used to increase the spareness of the model where it often leads to solutions where some elements of $\Lambda$ are exactly zero [Hastie et al., 2001]. This prior implies an independent double exponential (or Laplace) distribution for each parameter. The $l_1$-norm regularizer will induce sparseness and prune the parameter space and this is a desirable property for ACRFs optimization (see Chapter 6).

The $l_1$ and $l_2$ regularizers are convex in the parameters. Hence, this definitely maintains the convexity of the objective function. In addition, $l_1$ norm for vectors far from coordinate axes are larger than any $l_p$ norm for $p > 1$.

In a Bayesian setting, where all parameters are treated as random variables, maximizing the marginal likelihood can find regularized solutions via *model comparison* [Berger, 1985]. The marginal likelihood score can be obtained by adding a penalty term to the CML criterion known as Occam's or Bayes's factor. This term is a measure of the complexity of the model. This idea is explored and described in Chapter 4, with the incremental greedy forward model construction used to increase the sparseness of the final model.

# Chapter 4

# Nonparametric Entropy Classification

This chapter aims to delineate the details of an implementation of a conditional random field kernel machine, which is called the MaxEnt Kernel Machine (MEKM). The MEKM is a classification algorithm that aims to find nonlinear decision boundaries in an observation space. In particular, a low dimensional input space is mapped into a high dimensional kernel space. Hence, linear decision boundaries estimated by the MaxEnt principle implying minimization of the cross entropy between the data model and the hypothesized MaxEnt model, are used for classification.

In the next section, the basic idea behind the nonparametric classification is introduced with a focus on kernel spaces[1] and classification machines based on kernel methods. In Section 4.2, sparse kernel spaces are developed to reduce the computational complexity of constructing kernel spaces. The developed MEKM algorithm is presented in Section 4.3.1. The implementation details of the MEKM algorithm is based on a dimensionality reduction procedure and its parameter estimation is based on the adaptive approximate iterative scaling algorithm developed in Chapter 3. In

[1] Kernel spaces will be an abbreviation to the kernel induced feature spaces.

addition, the MEKM algorithm takes advantage of the probabilistic nature of the MaxEnt problem. Hence, its capacity control is implemented by treating the sparsity problem as a model selection problem to achieve high sparsity while maintaining low generalization errors. Finally, Section 4.4 reports experimental results for a vowel classification task.

## 4.1 Nonparametric Classification

The nonparametric classification problem involves finding or estimating decision boundaries or separating hyperplanes between classes or states $\mathbf{s}$ and observations $\mathbf{o}_t$ of an input space, where $\mathbf{o}_t \in \mathbb{R}^d$ and $t$ is the example index. Hence, the observation space can be divided into partitions or regions that can be assigned different labels according to the classification process. The empirical knowledge is collected via the noisy training data, $\mathcal{D} = \{(\mathbf{o}_t, \mathbf{s}_t)\}_{t=1}^{N}$ and there is no information available about the form of the function that defines the relationship between the states and the observed random process. On the other hand, parametric classification simplifies the estimation problem by making some assumptions related to the underlying model that generated data set $\mathcal{D}$ [Duda et al., 2000].

In general, nonparametric classification algorithm aims to find nonlinear decision boundaries in the observation space. This may be achieved by mapping a low dimensional input space into a high dimensional space and a linear classifier is trained for classification. As shown in Figure 4.1, moving to high dimensional spaces aims to simplify the classification problem as high dimensional spaces are more likely to be linearly separable than low dimensional spaces [Cover, 1965].

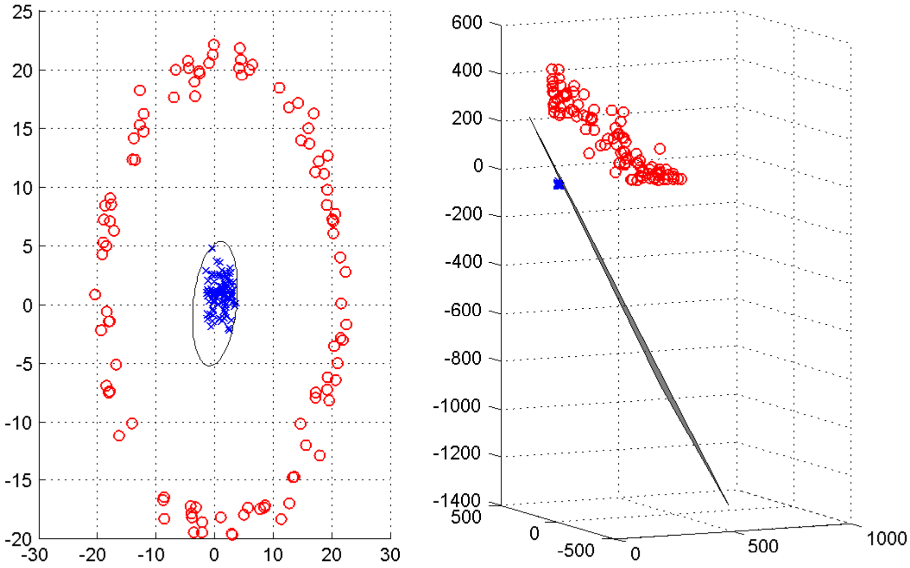


Figure 4.1: Two dimensional classification problem with nonlinear decision boundary is linearly separable in three dimensional space with a transformation function $\phi$: $\mathbb{R}^2 \rightarrow \mathbb{R}^3 = (\mathbf{o}_1, \mathbf{o}_2) \rightarrow (\mathbf{o}_1^2, \mathbf{o}_1\mathbf{o}_2, \mathbf{o}_2^2)$.

Several approaches have been developed to construct nonlinear classifiers. Multilayer perceptrons (MLPs) are feedforward neural networks that construct such classifiers using several hidden layers. MLPs can be trained with the standard backpropagation (Backprop) algorithm [Rumelhart et al., 1986]. The Backprop algorithm modifies the weights (i.e. hidden nodes parameters) of a hidden layer to minimize or maximize an objective function. In this work, we focus on feedforward neural networks that have only one hidden layer to construct a high dimensional space. On the other hand, the networks based on fixed or self-organized radial basis functions (RBF) are developed based on quadratic optimization [Powell, 1987]. GMMs may be used to construct the radial basis functions by clustering the data into a large number of Gaussians representing the dense regions of the observation space [Dempster et al., 1977]. Projecting the low dimensional data into high dimensional spaces in RBF and MLPs modelling is done by scoring a large number of hidden units (basis functions) representing the hidden layer nodes. The state-of-the-art classification algorithm, Support Vector Machine (SVM) introduced by Vapnik [Boser et al., 1992, Cortes and Vapnik, 1995, Vapnik, 1995, Vapnik, 1998] constructs a linear classifier in a high dimensional kernel space. The SVM training algorithm was designed to maximize the margin between two classes, which may lead to low generalization errors. The SVM algorithm can *automatically* specify the number of basis functions, and their centers, and the bias and linear weights by solving a quadratic programming problem. This may be the main reason behind the success of SVMs. In next section, kernel spaces, a key idea behind SVMs, will be reviewed in detail as they are related to the MEKM algorithm. The MEKM algorithm is similar to SVMs, since it uses kernel spaces to construct a high dimensional space.

### 4.1.1 Kernel Spaces

Nonparametric approaches based on the idea of reproducing kernel Hilbert spaces (RKHS) have been introduced in the field of pattern recognition [Aizerman et al., 1964, Boser et al., 1992] as well as in the field of function approximation [Aronszajin, 1950, Parzen, 1961, Wahba, 1990].

In feature spaces induced by Mercer's kernels [Boser et al., 1992], the original data (vectors) in the low dimensional space is projected via a nonlinear transformation into an infinite dimensional feature space. This is done by mapping the low dimensional input space into some other dot product space (i.e. known as *feature space*) via a nonlinear transformation $\phi$:

$$\phi : \mathbb{R}^d \to \mathbb{R}^N \tag{4.1.1}$$

where the problem of finding an appropriate nonlinearity $\phi$ is replaced by the problem of finding for an appropriate inner-product kernel function (i.e. $\phi$ is not calculated explicitly):

$$k(\mathbf{o}, \acute{\mathbf{o}}) = \langle(\phi(\mathbf{o}).\phi(\acute{\mathbf{o}})\rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{o})\phi_i(\acute{\mathbf{o}}) \tag{4.1.2}$$

where $\sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{o})\phi_i(\acute{\mathbf{o}})$ is an expansion of $k(\mathbf{o}, \acute{\mathbf{o}})$ in terms of the eigenfunctions, $\phi_i$, and the eigenvalues, $\lambda_i$. Mercer's theorem gives the necessary and sufficient conditions for a continuous symmetric kernel function, $k(\mathbf{o}, \acute{\mathbf{o}})$, to admit an expansion in terms of eigenfunctions and eigenvalues. Consequently, Mercer's kernels allow computations in possibly infinite dimensional feature spaces to be performed in finite dimensional kernel spaces. The kernel spaces dimensionality is related to the number of training data $N$. The $N$ x $N$ matrix $K$, called the kernel matrix, whose $ij$-th element is the inner-product kernel $k(\mathbf{o}_i, \mathbf{o}_j)$. A kernel function satisfies Mercer's theorem must generate a positive semidefinite matrix $K$ and any function is a valid kernel if it represents a dot product in some feature space. A common kernel function is the Gaussian (RBF) kernel and it is shown in equation[2] (4.1.3) ( for other kernel functions see [Scholkopf and Smola, 2002]).

$$k(\mathbf{o}, \acute{\mathbf{o}}; \sigma) = \exp\Big( - \frac{\|\mathbf{o} - \mathbf{o}'\|^2}{2\sigma^2} \Big) \tag{4.1.3}$$

Several techniques can be used to design new kernels [Scholkopf and Smola, 2002]. The Fisher kernel is an important technique to design new kernels for sequential processing [Jaakkola and Haussler, 1998]. A variable length sequence of observations is mapped into a vector defined in a space of fixed dimensionality, which is called score-space. For example, generative models such as HMMs are used to model patterns with variable length sequences and the gradient vector of the log-likelihood,

[2] Without loss of generality, the work in this chapter is formulated based on Gaussian kernels.

$\nabla_\Lambda \log p_\Lambda(\mathbf{O}|\mathcal{M})$, generates the image vector of the observation sequence, $\mathbf{O}$, in the score-space. The dimensionality of generated vectors is related to the number of the parameters, $\Lambda$, in the generative model. The dimensionality of the score-space is related to the number of the observation sequences. As a result, kernel based classification can be easily applied to classify patterns with variable length sequences. The score-space kernels [Smith et al., 2001, Smith and Gales, 2002], which is a generalization of the Fisher kernel, are used for post-processing in an HMM based speech recognition.

In general, kernel based spaces shift the modelling problem from *adapting* the parameters of the estimated models in the classification machine to *selection* of the most useful data points that represent the data, which implies compression of the observation space, while maintaining the discrimination capability to certain accuracy. In the next section, we will review the main classifiers developed based on kernel spaces.

### 4.1.2 Kernel Based Classification

The Support Vector Machine (SVM), the state-of-the-art classification algorithm, has been introduced by Vapnik [Boser et al., 1992, Cortes and Vapnik, 1995, Vapnik, 1995, Vapnik, 1998]. A tutorial on SVMs can be found in [Burges, 1998]. The SVM formulation is originally designed for binary classification and multiclass classification based on an SVM formulation is still an ongoing research.[3] The SVM formulation

[3] A formulation for k-class SVM is independently derived by [Vapnik, 1998] and [Weston and Watkins, 1998]. In Vapnik's one-against-all approach, each classifier is trained to sperate one class from the remaining k-1 classes. As a result, k classifiers are trained and the one with the highest score is used for classification [Vapnik, 1995]. Alternatively, the one-against-one approach trains all possible pairs of classes. Consequently, naive one-against-one classification may be computationally expensive. Another multiclass SVM approach is based on the principle of error-correcting

uses the training data in the form of inner-product. Hence, it takes advantage of kernel spaces to construct nonlinear decision boundaries in the original input space.

The SVM formulation is based on the structural risk minimization (SRM) principle, which minimizes an upper bound on the generalization error. This bound consists of two terms; the first one is the empirical risk and the second one is related to the capacity of the model. The *optimal* separating hyperplane, $\mathbf{w}^T\mathbf{o}+b$, separates two classes and maximizes the *margin* (i.e. distance to the closest point from either class). As a result, the SVM classifier maximizes the margin between two classes and minimizes the misclassification error on the training data. It can be shown that the optimal weight vector $\mathbf{w}$ having the minimum Euclidean norm is equivalent to maximizing the margin between two classes. Consequently, the SVM minimizes:

$$\mathcal{F}_{\text{SVM}}^{primal}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{4.1.4}$$

subject to the constraints,

$$\begin{aligned} \mathbf{w}^T\mathbf{o}_i + b &\geq +1 - \xi_i \quad \text{for } \mathbf{s}_i = +1 \\ \mathbf{w}^T\mathbf{o}_i + b &\leq -1 + \xi_i \quad \text{for } \mathbf{s}_i = -1 \\ \xi_i &\geq 0 \qquad\qquad \forall i \end{aligned} \tag{4.1.5}$$

where $\xi_i$ are called slack variables and are introduced to allow the margin constraints to be violated. This is done by solving a dual quadratic programming problem, to find the Lagrange multiplier $\{\lambda_i\}_{i=1}^{N}$ which maximizes

$$\mathcal{F}_{\text{SVM}}^{dual}(\Lambda) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j\mathbf{s}_i\mathbf{s}_j k(\mathbf{o}_i,\mathbf{o}_j) \tag{4.1.6}$$

subject to the constraints $\sum_{i=1}^{N}\lambda_i\mathbf{s}_i = 0$ and $0 \leq \lambda_i \leq C\ \forall i$. The parameter $C > 0$ is a regularization parameter and $\mathbf{s}_i \in \{-1,+1\}$. The training vectors $\mathbf{o}_i$ for which

codes and hamming distance decoding [Allwein et al., 2001], which is widely used in communication problems.

the $\lambda_i > 0$ are called *support vectors.* Support vectors consist of the vectors, which lie within or on the margin, or are misclassified. The SVM decision function is

$$f(\mathbf{o}) = \text{sign}\Big(\sum_{i=1}^{N} b + \lambda_i \mathbf{s}_i k(\mathbf{o}, \mathbf{o}_i)\Big) \tag{4.1.7}$$

It was shown that the SVM's maximum margin optimization can be formulated as a *hinge* Loss + Penalty criterion for a binary classification problem [Wahba, 1999]:

$$\sum_{i=1}^{N} L\Big(\mathbf{s}_i f(\mathbf{o_i})\Big) + \alpha \mathbf{w}^T \mathbf{w} = \sum_{i=1}^{N} [1 - \mathbf{s}_i f(\mathbf{o_i})]_+ + \alpha \mathbf{w}^T \mathbf{w} \tag{4.1.8}$$

where $f(\mathbf{o}) = \Big(\sum_{i=1}^{N} b + \lambda_i k(\mathbf{o}, \mathbf{o}_i)\Big)$, $[x]_+ = \max\{0, x\}$, and $\alpha = \frac{1}{2C}$. The hinge loss is a soft approximation to the misclassification error and leads to an explicit sparse solution (i.e. $\lambda_i = 0$ for some vector $i$). The solution is sparse because the hinge loss has zero values over an interval. In general, a loss function must be zero values over an interval in order to have a sparse expansion [Scholkopf and Smola, 2002].

SVM solution does not have a probabilistic interpretation since it just optimizes a discriminative objective function $\mathcal{F}_{\text{SVM}}(\Lambda)$. In order to estimate the posterior probabilities over the state, another probabilistic (MaxEnt) classifier has to be trained on the top of the output of a SVM classifier [Platt, 2000]. Alternatively, the classification problem can be directly formulated based on the MaxEnt principle in a high dimensional kernel space. This algorithm will lead to a kernel logistic regression (KLR) classifier and the import vector machine is a particular implementation of this algorithm [Zhu and Hastie, 2001]. Kernel logistic regression minimizes:

$$\mathcal{F}_{\text{KLR}}(\Lambda) = -\sum_{i=1}^{N} \mathbf{s}_i f(\mathbf{o}_i) - \ln(1 + \exp(f(\mathbf{o}_i))) + \alpha ||f||^2_{\mathcal{H}_k} \tag{4.1.9}$$

where $f(\mathbf{o}) = \Big(\sum_{i=1}^{N} b + \lambda_i k(\mathbf{o}, \mathbf{o}_i)\Big)$ and $\mathcal{H}_k$ is the RKHS generated by the kernel $k$. The penalty term $||f||^2_{\mathcal{H}_k} = \lambda^T K \lambda$ is added to the objective function, which is a norm calculated in $\mathcal{H}_k$ and $K$ is a positive definite matrix. The parameter $\alpha > 0$ is a regularization parameter and $\mathbf{s}_i \in \{0, +1\}$. KLR is a particular MaxEnt problem and its optimization can be done by methods detailed in Chapter 3. Equation(4.1.9) has a Loss + Penalty setup, where the loss is the KL divergence and the penalty term can be optional. KLR formulation does not lead to explicit sparse solutions ($\lambda_i \neq 0 \ \forall i$) and sparsity is achieved via shrinking methods.

Although $\mathcal{F}_{\mathrm{SVM}}(\Lambda)$ and $\mathcal{F}_{\mathrm{KLR}}(\Lambda)$ are motivated with different goals and setup, the two methods may have similar abstracted interpretation. In fact, two problems may be addressed after mapping the data into a high dimensional kernel space:

- How to construct the *linear* separating hyperplane in the constructed high dimensional space. The SVM utilizes the maximum margin principle, which tries to maximize the distance between the hyperplane and the closest data points of the two states. In KLR and MEKM methods, the decision boundary is estimated by the MaxEnt principle, which implies minimization of the cross entropy between the data model and the hypothesized MaxEnt model.

- Model complexity or sparsity, a measure of the observation space compression, which means how many data points can effectively represent the observation space without degrading the classification accuracy. Sparse solutions with few number of data points or support vectors are desirable when it is possible to achieve very low generalization errors.

As a result, this interpretation indicates that SVM and KLR/MEKM may lead to similar modelling ability given that they are based on kernel spaces. However, this thesis is based on probabilistic models and MEKM is a natural choice to construct kernel machines for classification.

## 4.2 Sparse Kernel Spaces

The dimensionality of a kernel space is $N$, where $N$ is the number of training examples and its construction time or storage complexity is $O(N^2)$. Hence, such spaces are limited to small scale problems. To overcome this problem, sparse kernel spaces are introduced for medium and large scale problems (i.e. $N \gg 10,000$).

The basic idea of the constructed spaces is that the inner product between two vectors $\mathbf{o}_i$, $\mathbf{o}_j$ will be very small for any vector $\mathbf{o}_j$ far from neighbours of vector $\mathbf{o}_i$. Hence, sparse kernel matrices can be obtained by pruning those elements that have very low scores. There is a relation between the inner product of two vectors and the Euclidean distance similarity measure and the idea is general for any $k(\mathbf{o}_i, \mathbf{o}_j)$. Approximate $k$ nearest neighbour search can efficiently construct such sparse spaces since exact $k$ nearest neighbour is not scalable for medium or large scale problems. Hence, the aim of sparse kernel spaces is to change the storage complexity of kernel spaces from $O(N^2)$ to $O(Nk)$, where $k$ is size of the nearest neighbor shortlist of elements and $k \ll N$. The search complexity of the approximate spaces will be estimated later in this section. Approximate search may lead to considerable reduction of the computation and storage required for large scale problems. In addition, pre-computed sparse kernel matrices can accelerate the incremental greedy feature induction (i.e. shrinking) algorithms as described in Section 4.3.2.

The $k$d-tree[4] is a multidimensional search tree data structure for sorting and organizing points in multidimensional spaces and it is a generalization of binary search in multidimensional spaces [Bentley, 1975, Omohundro, 1987, Moore, 1991]. $k$d-trees

[4] The letter $k$ in the word $k$d-tree refers to $k$-dimensional space. In this work, we refer to $k$ as nearest neighborhood shortlist elements and $d$ as the dimensionality of a space.

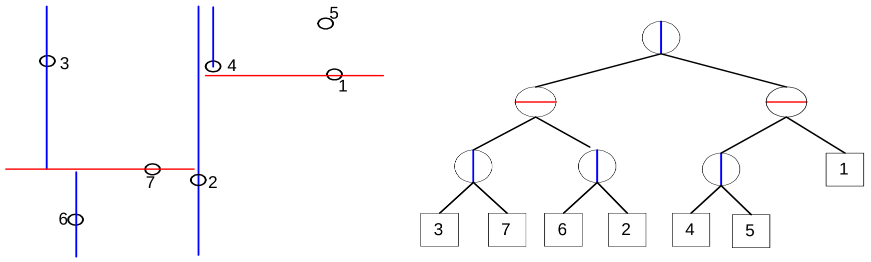


Figure 4.2: $k$d-tree storing the two dimensional location of a set of points. Each node represents an axis parallel split with points in leaves.

split planes that are perpendicular to the coordinate axes and the constructed partitions or regions are hyperrectangles as shown in Figure 4.2. The search time average complexity is $O(k\log N)$ for exact $k$ nearest neighbour search. In addition, the search cost is exponentially dependent on $d$, the dimensionality of the problem. Consequently, $k$d-trees are most effective data structures for problems with small and medium numbers of dimensions.

Since the time complexity of building a static tree is $O(N\log N)$, it may be desirable to keep $N$ small. To take advantage of this idea, a tree is constructed for each state $\mathbf{s}$ in the training data rather than one global tree for the whole data. Hence, the average complexity of search time is $O(|\mathbf{s}| \times k\log N_{\mathbf{s}})$, where $\sum_1^{|\mathbf{s}|} N_{\mathbf{s}} = N$. This process will generate a $|\mathbf{s}| \times k$-shortlist. Based on the calculated distances for each element in the $k$-shortlist, $|\mathbf{s}| \times k$-shortlist elements are resorted and only the top $k$-shortlist are selected. This cheap resorting operation $O(|\mathbf{s}| \times k)$ yields a minor change to the overall search time.

Traversing each state tree during the search process may be inefficient if the system has a large number of states. Hence, it may be important to *rank* the trees during the search for a target point $\mathbf{o}_i$ and only traverse trees that are most relevant to the target point during the approximate neighborhood search. A simple and efficient algorithm is proposed to evaluate the rank of each tree based on a Gaussian model approximation. Mean and covariance for each tree's observation data are calculated and the constructed Gaussian model is associated with each tree. For each target point, the estimated Gaussians representing the trees are evaluated and the likelihood scores are taken as rank of trees. Only the top $r$ trees are traversed during the search process, where $r \ll |\mathbf{s}|$. This will change the overall search time to $O(|\mathbf{s}| + r \times k \log N_r)$. An abstracted interpretation of this idea is that the selected points from this approximate search focuses only on confusable states for a target point during the modelling process in next stages. Note that, the ranking process leads a stochastic and approximate $k$ nearest neighbour search.

This pruning algorithm can run offline to save the sparse kernel matrix to a disk before the actual modelling process. The algorithm may construct spaces based on a large number of data points and the quality of the constructed spaces are a function of $k$, which balances the approximation quality and the storage complexity. Moreover, $r$ balances approximation quality and search time required to find the $k$ elements.

One approach to handle large scale optimization based on kernel spaces within the context of SVM optimization, is to decompose large scale problems into a series of smaller problems [Osuna et al., 1997, Joachims, 1998]. Hence, a kernel matrix representing a small amount of the training data is cached and loaded into memory to solve a small classification problem. Alternatively, sparse kernel spaces take advantage of the computational geometry of kernel spaces to handle large scale problems and it aims to minimize the computations and memory usage. Hence, decomposition algorithms and the purposed sparse computational geometry algorithm can complement each other to handle large scale problems based on kernel methods.

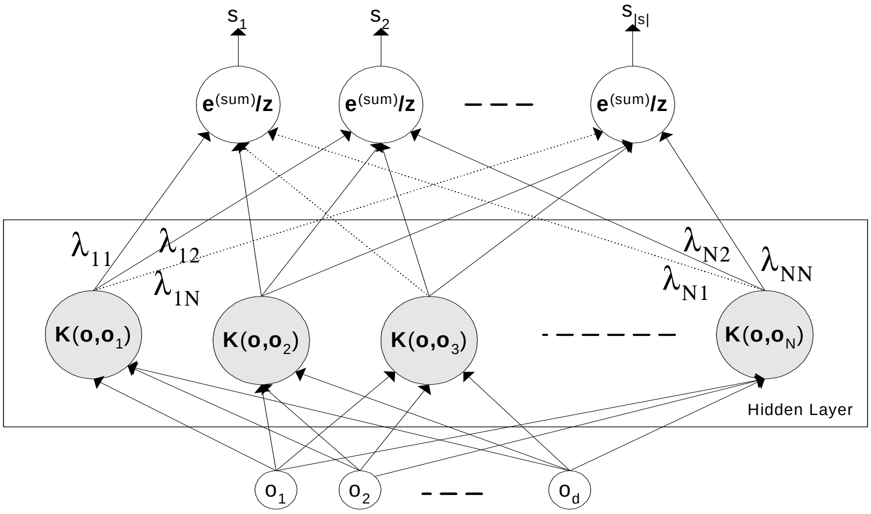


Figure 4.3: The proposed architecture of the MEKM algorithm. The network has one hidden layer of inner-product kernels constructed from the input data. The hidden layer is connected to the output layer via sparse connections. The output layer estimates the posterior probabilities over the states.

In general, kernel spaces do not focus on dense regions in the observation spaces and they construct high dimensional spaces in a blind way, without integrating some useful prior information. Sometimes, it is computationally more efficient to locate these dense regions with arbitrary resolution in advance and construct spaces with dimensionality $\ll N$. Such spaces are proposed and detailed in Chapter 6 and they are more efficient to handle speech recognition applications.

## 4.3 The Maximum Entropy Kernel Machine

The MEKM algorithm, which is a conditional random field based on kernel spaces is shown in Figure 4.3. The MEKM architecture can be understood as a one hidden layer neural network. The number of hidden nodes theoretically equals to the number of data points $N$ but it is usually small (i.e. $\ll N$).

The MEKM algorithm can be used to construct either binary or multiclass classifiers. The estimated model by the MEKM algorithm is given by

$$P_\Lambda(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_\Lambda(\mathbf{o})} \exp\Big(\sum_i \lambda_i g_i(\mathbf{o}, \mathbf{s}; \acute{\mathbf{o}})\Big) \tag{4.3.1}$$

where the characterizing constraints $g_i(\mathbf{o}, \mathbf{s}; \acute{\mathbf{o}}) = \delta(\mathbf{s}, \mathbf{s}')k(\mathbf{o}, \acute{\mathbf{o}})$ equal to the kernel function score of an observation point **o** given the kernel point **ó**.

MEKM optimizes the KL divergence (Loss) between the data model and an hypothesized kernel based MaxEnt model. The penalty term is based on Occam's razor principle and it is detailed in Section 4.3.3. In the next section, MEKM implementation is presented and it is similar to the import vector machine [Zhu and Hastie, 2001]. The basic idea behind the training algorithm may be based on the feature induction algorithm developed in [Della Pietra et al., 1997].

### 4.3.1 MEKM Estimation

The modelling process involved in the MEKM algorithm is detailed in algorithm 4.1. Although the MaxEnt induction process is described for kernel spaces, it is a general procedure for any constraint formulation (see Section 3.3). For example, a feature selection algorithm was described in a previous work based on the same induction process [Hifny et al., 2004].

The implementation of the MEKM algorithm is similar to the import vector machine [Zhu and Hastie, 2001], because the two algorithms are based on kernel spaces and the estimated models are based on the MaxEnt principle. In addition, the two algorithms use an incremental greedy procedure to increase the CML objective function and the complexity of the constructed model. However, there are different details

**Algorithm 4.1** The Maximum Entropy Kernel Machine

```
1: Start with the *final model*, where no constraints are added (uniform model).
2: Define the *evaluated list* of constraints $g_i$, $i \in \{1, \dots, n\}$.
3: For each constraint in the evaluated list, compute the gain for each $g_i$ with respect to the final model (see Section 4.3.2).
4: Select the constraints that have the maximal gain $\triangle\mathcal{L}_{g_i}$ after sorting the constraints according to the their gains (i.e. $n$-best shortlist).
5: Construct the final model by adding the constraints, which yields the greatest gains.
6: Refine the model parameters using the adaptive AIS algorithm. {optional step}
7: Remove the constraints that have the maximal gains from the evaluated list.
8: If a valid condition is achieved, then stop else go to step 3 (see Section 4.3.3).
9: Refine the constructed MEKM model using the adaptive AIS for few iterations.
```

of the implementation since the MEKM is designed for large scale problems. These differences are:

- MEKM implementation takes advantage of the computational geometry of kernel spaces to prune the kernel spaces via nearest neighborhood methods. The sparse kernel spaces were detailed in Section 4.2 and they may be useful for medium or large scale problems.

- During the induction process, the gains of the evaluated constraints are calculated based on a first order approximation of the objective function in the MEKM implementation. This approximation is very efficient and faster than the second order Newton-Raphson's approximation used to calculate the gains in the import vector machine. The induction process is detailed in Section 4.3.2.

- MEKM models are trained using the adaptive AIS algorithm. This method is scalable for large scale problems $O(\#\text{parameters})$ and may achieve a good generalization accuracy as described in Chapter 3. On the other hand, the import vector machine is based on the exact Newton's method.

- Complexity control is different in the two implementations. The MEKM complexity control is measured based on a model selection criterion as detailed in Section 4.3.3. The import vector machine adds a penalty term $||f||^2_{\mathcal{H}_k}$ to the objective function as described in Section 4.1.2.

### 4.3.2 MEKM Dimensionality Reduction

In the MaxEnt solution, the Lagrange multipliers $\Lambda = \{\lambda_i\}$, which may be interpreted as the importance of each constraint, are the outcome of the training procedure. Kernel spaces result in MaxEnt problems with dimensionality related to the number of data points $N$. Hence, the estimation of the MaxEnt model parameters $O(\mathbf{s}N)$ may be computationally intensive and it will be a function of all kernels or data points (i.e. similar to instance/memory based learning). Obtaining the MaxEnt solution in incremental steps is a practical way to evaluate the importance of the constraints. In particular, this methodology may be considered as a form of dimensionality reduction and selection. Unfortunately, evaluating the importance of every constraint by building a MaxEnt model incrementally will invalidate the previous estimate of the model parameters. As a result, all the model parameters would have to be re-estimated at each step, which would be computationally intensive.

Della Pietra et al [Della Pietra et al., 1997] developed an efficient solution to the problem in which the Lagrange multipliers of the constraints are kept fixed while evaluating a given constraint (known as mean field approximation in statistical physics) as shown in the equation (4.3.2).

$$\hat{P}_{g_i}(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_{g_i}(\mathbf{o})} P_\Lambda(\mathbf{s} \mid \mathbf{o}) \exp\big(\beta_i g_i(\mathbf{o}, \mathbf{s})\big) \tag{4.3.2}$$

This yields a great computational saving as the problem is reduced to a one dimensional optimization problem. This inductive approach is based on a measure referred to as the constraint gain, where the gain of a constraint is usually measured by adding the evaluated constraint to the model and calculating the gain in terms of the reduction of entropy or equivalently the increase of the log-likelihood of the training data with respect to the model:

$$\begin{aligned} \mathrm{G}_{g_i}(\Lambda, \beta_i) &= \triangle\mathcal{L}_{g_i}(\Lambda, \beta_i) \\ &\approx \mathcal{L}(\hat{P}_{g_i}(\mathbf{s} \mid \mathbf{o})) - \mathcal{L}(P_\Lambda(\mathbf{s} \mid \mathbf{o})) \\ &\approx \sum_{\mathbf{o}} \tilde{p}(\mathbf{o})\log Z_{g_i}(\mathbf{o}) + \beta_i \tilde{p}(g_i) \end{aligned} \tag{4.3.3}$$

where $\hat{P}_{g_i}(\mathbf{s} \mid \mathbf{o})$ is an approximate MaxEnt model constructed after adding the constraint $g_i$ and $\beta_i$ is the estimated Lagrange multiplier for an evaluated constraint associated with the function $g_i(\mathbf{o}, \mathbf{s})$. $\mathcal{L}(\hat{P}_{g_i}(\mathbf{s} \mid \mathbf{o}))$ is the estimated log-likelihood of the MaxEnt model with respect to the training data *after* considering the constraint $g_i$. $\mathcal{L}(P_\Lambda(\mathbf{s} \mid \mathbf{o}))$ is the estimated log-likelihood of the MaxEnt model with respect to the training data *before* considering the constraint $g_i$. Since the evaluated constraints are not part of the current model, they are called the *inactive* set of constraints.

Equation (4.3.3) calculates the optimal gain associated with a constraint numerically using a one dimensional root-finding method like the Newton-Raphson method (i.e. no closed form solution). The Newton-Raphson method implies inner loops over the training data to find the optimal $\beta_i$. As a result, this method is not computationally efficient. The Newton-Raphson solution can be approximated using only one iteration and starting the solution from $\beta_i = 0$ [Zhu and Hastie, 2001].

An alternative method to calculate the gains of the inactive set of constraints is given by

$$
\begin{aligned}
\mathrm{G}_{g_i}(\Lambda, \beta_i) &= \eta \mathbf{g}(\Lambda, \beta_i)\Big|_{\beta_i=0} \\
&\propto |\tilde{p}(g_i) - \tilde{p}_{\Lambda,\beta_i}(g_i)|\Big|_{\beta_i=0}
\end{aligned}
\tag{4.3.4}
$$

Equation (4.3.4) gives the *approximate gain* of the inactive constraints with respect to the current model [Perkins et al., 2003, Lafferty et al., 2004]. It can be explained by expanding the objective function using Taylor's series and selecting only the first order approximation around the current solution. Hence, inactive set constraints that produce large gradient values are the most useful constraints for the modelling process. In fact, selecting or updating the parameters that have large gradient values can be a general rule to evaluate parameters for any objective function. This is due to the fact that the directions of the largest gradients are the most useful directions to increase/decrease an objective function. Constraint evaluation based on this gradient test is related to the optimization of the $l_1$-ACRF models (see Chapter 6).

The evaluated constraints are sorted and some of them ($n$-best) are added to the current model (i.e. *active* set of constraints) and are removed from the inactive set. The evaluated process needs one iteration over the training corpus and this overhead cannot be avoided. However, this penalty can be useful to get approximate initial values of $\beta_i$ in parallel to the evaluation process using an exact IS update, where $\beta_i$ can be estimated by

$$\lambda_i^{(\tau)} = \beta_i = \frac{1}{C} \log \frac{\tilde{p}(g_i|\mathbf{o})}{P_\Lambda(g_i|\mathbf{o})} \tag{4.3.5}$$

Equation (4.3.5) assumes that the kernel values are positive and their maximum values are one and this is the case for RBF kernels. Selecting the best constraint only (i.e. $n = 1$) provides a fast estimate, $C \approx 1.0$, for the parameters without the additional training cost of refining all the model parameters together after adding the constraints to the model. As a result, for small tasks, $n = 1$ is desirable and leads to the best sparsity. In contrast, for large scale problems, $n \gg 1$ can lead to a fast modelling process with some compromise with the sparsity of the model. This is due to the fact that the $n$-best constraints are correlated and adding them to the current model may lead to adding some redundant constraints during the induction process.

All the MaxEnt model parameters can be further refined by the adaptive AIS algorithm for few iterations, which is useful for large scale problems. Concurrent evaluation of constraints and model training progresses until a valid termination condition. In this work, we formulate the stopping condition as a model selection process and it is detailed in the next section.

### 4.3.3 MEKM Complexity Control

MEKM is basically a forward greedy algorithm that increases the CML criterion and the model complexity with respect to the data at each step. In the previous section, a criterion to select the most effective data points to present the whole observation space was presented. With a statistical measure for model complexity, the MEKM algorithm can stop the process of adding extra data points, given the measure of the complexity. Models with low complexity usually imply sparse solutions. The MEKM algorithm utilizes a Bayesian approach for measuring the model complexity, while the evolution of the forward greedy procedure.

The Bayesian approach adds a penalty term to the CML criterion to penalize the number of the parameters in the constructed model. This penalty term is known as the Bayes or Occam factor. In general, one would select simple models according to the Occam's razor principle, which is a particular case of structural risk minimization principle [Cortes and Vapnik, 1995]. Occam's razor principle[5] is often called the principle of parsimony and it may be stated as follows:

> Entities should not be multiplied beyond necessity.

Suppose we have a set of candidate models $\mathcal{M}_m$, $m = 1, \ldots, M$ and corresponding model parameters $\hat{\Theta}_m$. The MEKM algorithm generates such a set of models through the forward greedy algorithm. Each step of the forward greedy algorithm (e.g. $m$) generates the model $\mathcal{M}_m$ and its complexity is related to $\hat{\Theta}_m$. Using Bayes theorem, the posterior probability of the $\mathcal{M}_m$ models is

$$P(\mathcal{M}_m \mid \mathcal{D}) \propto p(\mathcal{D} \mid \mathcal{M}_m) P(\mathcal{M}_m) \tag{4.3.6}$$

and if the model prior $P(\mathcal{M}_m)$ is uninformative, then $P(\mathcal{M}_m \mid \mathcal{D}) \propto p(\mathcal{D} \mid \mathcal{M}_m)$. The marginal likelihood or evidence $p(\mathcal{D} \mid \mathcal{M}_m)$ is given by

$$p(\mathcal{D} \mid \mathcal{M}_m) = \int p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) p(\hat{\Theta}_m \mid \mathcal{M}_m) \, \mathrm{d}\hat{\Theta}_m \tag{4.3.7}$$

and the posterior probability of the parameters is given by

$$p(\hat{\Theta}_m \mid \mathcal{D}, \mathcal{M}_m) = \frac{p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) p(\hat{\Theta}_m \mid \mathcal{M}_m)}{p(\mathcal{D} \mid \mathcal{M}_m)} \tag{4.3.8}$$

At each step of the greedy forward procedure, after the CML estimation of the model parameters, the marginal likelihood can be locally approximated by the *Laplace Approximation* [Berger, 1985, MacKay, 2003] as

$$\begin{aligned} \log p_{\mathrm{LA}}(\mathcal{D} \mid \mathcal{M}_m) \approx \log p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) - \frac{|\hat{\Theta}_m|}{2} \log N + \log p(\hat{\Theta}_m \mid \mathcal{M}_m) \\ + \frac{|\hat{\Theta}_m|}{2} \log 2\pi - \frac{1}{2} \log |\frac{1}{N} \mathbf{H}(\hat{\Theta}_m)| \end{aligned} \tag{4.3.9}$$

[5] It is related to Albert Einstein's quote: "Make everything as simple as possible, but not simpler."

where $\mathbf{H} = -\nabla\nabla \log P(\hat{\Theta}_m \mid \mathcal{D}, \mathcal{M}_m)$, is the Hessian matrix of the second derivative of the negative log posterior at the CML mode. The two last terms of the Laplace approximation can be dropped as they are asymptotically dominated by the other terms. On the other hand, the model parameters priors $P(\hat{\Theta}_m \mid \mathcal{M}_m)$ are assumed to be uninformative. Hence, the evidence can be approximated by

$$\log p_{\text{BIC}}(\mathcal{D} \mid \mathcal{M}_m) \approx \log p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) - \frac{1}{2}|\hat{\Theta}_m| \log N \tag{4.3.10}$$

and this is known as the *Bayesian information criterion* (BIC) [Schwarz, 1978]. Due to the first order approximation of the marginal likelihood, the BIC tends to penalize complex models more heavily, giving preference to simple models in the selection process (i.e. underfitting the true model). Hence, a scaling factor $\alpha$ is introduced in order to relax the conservative nature of the BIC model selection method:

$$\log p_{\text{BIC}}(\mathcal{D} \mid \mathcal{M}_m) \approx \log p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) - \alpha\frac{1}{2}|\hat{\Theta}_m| \log N \tag{4.3.11}$$

This scaling factor allows a simple way to impose a trade off between sparsity and modelling quality. The BIC score is used in some speech applications such as speaker segmentation [Tritschler and Gopinath, 1999] and decision tree state tying methods for HMM based speech recognition systems [Chou and Reichl, 1999].

The BIC score is related to Rissanen's *minimum description length* (MDL) criterion [Rissanen, 2005]. Another model selection method is based on Akaike's *information criteria* (AIC) [Akaike, 1974] and it is given by

$$\log p_{\text{AIC}}(\mathcal{D} \mid \mathcal{M}_m) \approx \log p(\mathcal{D} \mid \hat{\Theta}_m, \mathcal{M}_m) - |\hat{\Theta}_m| \tag{4.3.12}$$

The $|\hat{\Theta}_m|$ is measured as the summation of the number of the MaxEnt model parameters $|\Lambda_m|$ and the number of activated kernels $|a_m|$ in the model multiplied by the observation space dimensionality $d$ as shown in equation (4.3.13).

$$|\hat{\Theta}_m| = |\Lambda_m| + |a_m| * d \tag{4.3.13}$$

At each induction step of the modelling process, $P(\mathcal{M}_m \mid \mathcal{D})$ is approximated by the BIC score calculated for each generated model and compared to the previous model BIC score. The induction process is stopped when the BIC score starts to decrease. As a result, the Laplace approximation score or the BIC score is used as an evaluation condition to stop the kernel induction process.

## 4.4 Experimental Work

Vowel classification using Deterding database is a common task to evaluate new classification methods in speech recognition [Robinson, 1989]. The Deterding database was developed in the context of speaker normalization research [Deterding, 1989]. It is recorded of 11 British English vowels (i.e. 11 states). It consists of 528 samples in the training data and 462 samples in the test data. Each sample consists of a 10 dimensional acoustic vector (i.e. $d = 10$) derived from linear prediction coding (LPC) analysis. The training data and test data features are pre-processed to have zero mean and unit variance.

As shown in Table 4.1, Induction ITR is the number of iterations required to refine the models using the adaptive AIS algorithm (default) or AIS-mean during the induction process. In general, it is desirable to keep this number small in order to have a fast induction process. The count of import vectors (IV) selected by the algorithm (i.e. $m =$ IVs count) is a measure of the sparsity of a MEKM model. The BIC weight $\alpha$ controls the complexity of the models (i.e. how many IVs are active in

Table 4.1: *Vowel classification results using the MEKM algorithm.*

| $k$ | $\alpha$ | Induction ITR | IV | CA |
|---|---|---|---|---|
| 100 | 0.1 | 0 | 52 | 43.0% |
| 100 | 0.1 | 5 | 41 | 60.2% |
| 100 | 0.1 | 3 | 53 | **63.4**% |
| 200 | 0.01 | AIS-mean (10) | 252 | 61.9% |
| 200 | 0.01 | AIS-mean (20) | 192 | 62.7% |
| 200 | 0.01 | 0 | 67 | 50.6% |
| 200 | 0.01 | 5 | 194 | 60.6% |
| 200 | 0.01 | 10 | 216 | 59.0% |
| 200 | 0.01 | 20 | 197 | **64.9**% |

a model). $k$ is the parameter that controls the quality of the neighborhood search to construct the sparse kernel spaces. The other parameters are fixed to default values as follow RBF kernel variance $\sigma = 2$, confusability control parameter $r = 11$, and the number of selected constraints $n = 1$ during the modelling process. All the models are refined using 20 iterations using adaptive AIS algorithm, after the induction process is finished.

When the induction ITR number is zero, the selected constraint parameter is just updated using equation (4.3.5) and the model parameters are not refined after each induction step. Unfortunately, as shown in Table 4.1, the models estimated with this approach do not have a good generalization performance. Hence, training models after each induction step is important to improve the classifier performance. In general, for simple setup with $\alpha = 0.1$ and $k = 100$, it was possible to achieve a good accuracy (63.4 %) with 53 import vectors. In order to get better results, the problem complexity was increased with $\alpha = 0.01$, $k = 200$, and induction ITR (i.e. 20 iterations). A MEKM classifier with 197 import vectors, was able to achieve (64.9%).

The reported results (63.4 % or 64.9%) outperform some published results on this task detailed in [Hochreiter and Schmidhuber, 1999] and the maximum margin training of generative kernels (61.2 %) [Layton, 2004]. A state-of-the-art result based on one-against-one SVM training (70%) was reported in [Clarkson and Moreno, 1999]. However, one-against-one training needs to train $\mathbf{s}(\mathbf{s}-1)/2$ classifiers. On the other hand, one-against-one training needs special classification and voting algorithms to find the winner state [Kreel, 1998, Platt et al., 2000].

Since the SVM's publication did not report the sparsity properties of the solution (i.e. how many support vectors are active in the modelling process), LIBSVM a state-of-the-art SVM one-against-one classifier was trained on the same task [Chang and Lin, 2001]. The classification results using the LIBSVM is (62.1%) and the number of support vectors is 384, where the LIBSVM hyperparameters $\gamma = 2.0$ and $C = 8.0$ were chosen using its default cross validation procedure. Hence, MEKM was able to achieve a similar classification accuracy, with better sparsity properties (few import vectors). Different sparsity properties between SVM and MEKM may be related to that one-against-one training is not an elegant method to achieve good sparse solutions (i.e. it may behave like an intelligent $k$ Nearest Neighbour algorithm).

In this chapter, it was shown that a simple MaxEnt classifier (MEKM) based on kernel spaces leads to state-of-the-art results on a standard task. The success behind MEKM algorithm may be directly related to the use of high dimensional spaces. This may suggest that moving to high dimensional spaces can be a principled approach to improve speech recognition. As a result, a high dimensional space, which is computationally efficient was developed and integrated within the $l_1$-ACRF framework (see Chapter 6).

# Chapter 5

# Sequential Conditional Random Field Models

Hidden Markov Models are the choice for state-of-the-art stochastic speech recognition engines. The *inference* problem (i.e. the calculation of the probabilities of the state given the entire observation sequence) is tractable. Based on dynamic programming methods, this problem is solved by the forward-backward (Baum-Welch) algorithm as described in Chapter 2. Hence, HMM evaluation and training are computationally very efficient. As a result, HMMs are very popular graphical models for sequential modelling. HMMs are known to be a special case of Dynamic Bayesian Networks (DBNs) [Smyth et al., 1997].

Recently, there has been a strong interest in acoustic modelling formulations that do not make assumptions about the shape of the stochastic process generation (i.e. nonparametric models).[1] Hybrids of HMM/ANN have been proposed as an alternative for shape-based generative models [Morgan and Bourlard, 1995, Trentin and Gori, 2001]. We also formulated a variant of such methods based on

[1] Since all models have parameters, categorizing models as either parametric or nonparametric may be inaccurate. As a parametric model often has a form or shape (e.g. the Gaussian distribution is a bell shaped model), models are categorized as either shape/shapeless in this work.

the maximum entropy principle [Hifny et al., 2005]. Such models use a wide range of nonparametric classifiers and the parameter estimation problem usually has different details.

Based on the MaxEnt principle, sequential CRF models based on the first order Markov assumption have recently been introduced as a framework for sequence analysis [Lafferty et al., 2001]. The linear Markov chain CRF[2] is an undirected graphical model that can be thought of as the twin of an HMM as shown in Figure 5.1. CRFs do not assume a shape for the generation of the data but have global normalization with the partition function score $Z_\Lambda(\mathbf{O})$, which is similar to the total probability $p(\mathbf{O}|\mathcal{M})$ in the HMM framework as described in Chapter 2. The efficient calculation of the partition function or the total probability using the forward algorithm, is one of the main attributes behind the success of the HMMs and CRFs. Hence, CRFs are an appropriate choice for modelling sequences of observations, which provide conditional distributions to model the given data. Moreover, CRFs are flexible and scalable in describing the relationship between the observation sequences. They can be easily extended to higher order sufficient statistics, which are not commonly used in speech recognition applications.

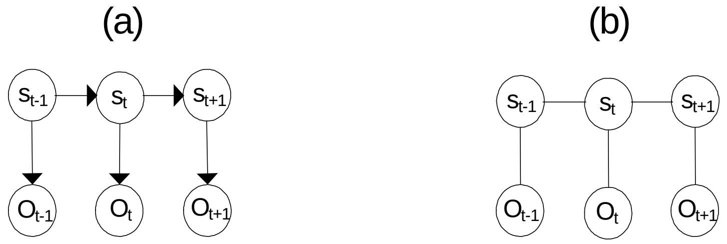


Figure 5.1: (a) an HMM, (b) an analogous linear chain CRF for phone representation.

[2] Graphical models based on CRFs have many varieties such as 2D-CRFs used for image applications. In this chapter and the next chapter, CRF will be used to refer to linear Markov chain CRF model.

CRFs are theoretically principled models for relaxing the HMM conditional independence assumption. This is due to the fact that these models are based on potential functions or constraints, which may have correlated information and the constraints can be statistically dependent. Hence, using $\Delta$ and $\Delta\Delta$ features, or acoustic context information from correlated frames as constraints within the CRF framework, is theoretically justified. Note that, using $\Delta$ and $\Delta\Delta$ features within the HMM framework is theoretically inaccurate but pragmatically it improves the recognition performance.

In general, all acoustic models developed based on the MaxEnt principle can be considered as CRFs since they have the same root but with different graphical relationships. In addition, HMMs trained by relaxing the probabilistic constraints during the training process (i.e. unconstrained parameters) may be considered as undirected graph CRF models [Kapadia, 1998]. Hidden neural networks were motivated within undirected graphical modelling and global normalization was provided [Krogh and Riis, 1999]. Examples of CRFs introduced for acoustic modelling are [Likhododev and Gao, 2002, Macherey and Ney, 2003, Hifny et al., 2005, Hifny and Renals, 2005, Gunawardana et al., 2005].

Acoustic models such as direct models[3] [Gao, 2003] or hidden conditional random fields [Gunawardana et al., 2005] investigate the estimation of the language model parameters along with the acoustic model parameters forming a unified model. In this work, we do not train the language model and the acoustic model as a single model. This is due to the basic fact that estimating the language model parameters using $n$-gram methods and linear interpolation is very efficient. In addition, training language model parameters based on CRF/MaxEnt models did not lead to

[3] In addition to acoustic constraints, the direct models approach trains linguistic and semantic constraints within a unified framework.

significant improvement over the conventional $n$-gram and linear interpolation methods [Rosenfeld, 1994]. Hence, the acoustic modelling problem is separated from the language modelling problem, as is done in current HMM based speech recognition systems. Separating language model parameter estimation from the acoustic modelling problem is desirable to make the parameter estimation problem less complex. In addition, this approach will allow us to reuse many components of the current speech recognizers without any implementation change. Thus, direct evaluation of acoustic models based on CRFs with respect to HMM acoustic models can be addressed by fixing the language model parameters. This will lead to the nonparametric CRF twin of the HMM models, as shown in Figure 5.1.

This chapter will concentrate on acoustic model based CRFs, which are identical to HMMs, while fixing the language models parameters. We will describe a discriminative training scheme aimed at maximizing the posterior probability of the correct word (phone) sequence given the acoustic observations, which is conceptually very similar to the discriminative training methods for HMM models described in Chapter 2. In the next chapter, another family of CRF acoustic models is introduced, which uses explicit acoustic context information in an augmented high dimensional space representation.

In the next section, we give a short introduction to Conditional Random Fields and present our motivation for using them as the basic models for speech recognition. Section 5.2 discusses and reviews exponential family distributions and establishes connections between HMMs and CRFs, which are useful for a practical implementation for CRFs. In Section 5.3, a brief overview on the discriminative training procedure of the CRFs is presented. In addition, a new Approximate Iterative Scaling algorithm for training CRFs is presented. The new algorithm is compared with gradient based optimization. Section 5.4 reports experimental results for a phoneme recognition task using the TIMIT database.

## 5.1 Conditional Random Fields

The maximum entropy formalism for sequential modelling results in a probability distribution, which is the log linear or exponential model:

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\Big(\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t, t) + \sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t)\Big) \qquad (5.1.1)$$

where

- $P_\Lambda(\mathbf{S}|\mathbf{O})$ obeys the *Markovian* property $P_\Lambda(\mathbf{s}_t|\{\mathbf{s}_\tau\}_{\tau\neq t}, \mathbf{O}) = P_\Lambda(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{O})$.
- $\lambda^i_{\mathbf{s}_t}$ and $\lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}}$ are the Lagrange multiplier (weighting factors) associated to the characterizing functions $b_i(\mathbf{O}, \mathbf{s}_t, t)$ and $a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t)$.
- $Z_\Lambda(\mathbf{O})$ (Zustandsumme) is a normalization coefficient resulting from the natural constraints over the probabilities summation, commonly called the partition function and given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}} \prod_{t=1}^{T} \exp\Big(\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t, t) + \sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t)\Big)$$

The partition function $Z_\Lambda(\mathbf{O})$ score is similar to the total probability $p(\mathbf{O}|\mathcal{M})$ score in HMM modelling, which is described in Chapter 2. The conditional distribution behind the CRF model as shown in equation (5.1.1) implies arbitrary combinations of state scores $b_i(\mathbf{O}, \mathbf{s}_t, t)$ and transition scores $a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t)$. Hence, it is conceptually similar to HMMs that have *only* two scores; emission probability $p(\mathbf{o}_t|\mathbf{s}_t)$ and transition probabilities $P(\mathbf{s}_t|\mathbf{s}_{t-1})$. CRFs offer a principled framework for combining different state scores in a natural way. The HMMs and CRFs share the first order Markov assumption, which simplifies the training and decoding algorithms.

CRFs have an attractive property: the MaxEnt models (linear chain CRFs are a special case) make little assumptions, as they are the most unbiased distributions that are simultaneously consistent with a set of constraints. Hence, CRF models do not suffer from the observation independence assumption made in the HMM framework, as the characterizing functions may be statistically dependent or correlated. This is very clear in the model equation where the characterizing functions $b_i(\mathbf{O}, \mathbf{s}_t, t)$ are arbitrary functions over the *entire* observation sequence $\mathbf{O}$. Moreover, CRF models do not constrain the shape of the data generation and the modelling quality is a function of the sufficient statistics represented by the characterizing functions. In speech recognition problems, second order sufficient statistics are extracted from the acoustic observations.

The state characterizing function $b_i(\mathbf{O}, \mathbf{s}_t, t)$ can depend only on the current observation (i.e. observation $b_i(\mathbf{O}, \mathbf{s}_t, t) = b_i(\mathbf{o}_t, \mathbf{s}_t, t)$). For example, front end speech processing generally extracts MFCC+$\Delta$+$\Delta\Delta$ as the basic acoustic vector , the observation dependent term in equation (5.1.1) is given by

$$\begin{aligned}\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t, t) =& \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{o}_t, \mathbf{s}_t, t) \\ =& \lambda^0_{\mathbf{s}_t} b_0 + \sum_{i=1}^{2d} \Big( \lambda^i_{\mathbf{s}_t} \mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta\mathbf{o}_{ti} \\ & + \lambda^i_{\mathbf{s}_t} \mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta\mathbf{o}^2_{ti} \Big)\end{aligned} \tag{5.1.2}$$

where $b_0 = 1$ is the bias constraint, $d$ is the vector dimensionality, and $\mathbf{o}_{ti}$, $\mathbf{o}_{ti}^2$ are the first and second order moments of the acoustic features. In addition, with *one* transition characterizing function, the transition dependent term in equation (5.1.1) is given by

$$\sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t) = \lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) \tag{5.1.3}$$

where $a(\mathbf{s}_t, \mathbf{s}_{t-1}, t)$ is a binary feature and $\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}}$ is related to $\log a_{\mathbf{s}_t\mathbf{s}_{t-1}}$ in HMM modelling.

Equations (5.1.2) and (5.1.3) will lead to a nonparametric identical formulation similar to an HMM (see Figure 5.1). The state characterizing function $b_i(\mathbf{O}, \mathbf{s}_t, t)$ can depend on neighboring observations as shown in Figure 6.2. CRF models that take advantage of the neighborhood acoustic observations are described in a previous work, where the neighboring observations are utilized to model the acoustic context in an augmented high dimensional space [Hifny and Renals, 2005]. The CRF models based on augmented spaces will be detailed in Chapter 6.

## 5.2 Exponential Family

Like HMMs, CRF models can be trained by a generative training procedure which maximizes the likelihood between the data and the underlying distributions (i.e. no decision boundaries are estimated). The latent variable CRFs and more generally the latent MaxEnt principle are addressed in [Wang et al., 2002], where the authors estimate mixture models up to second order sufficient statistics. They described a two step EM-GIS algorithm[4] to train the models where the state sequence is not known.

[4] A two step EM-GIS algorithm forms a Generalized EM (GEM) algorithm. Using a GEM algorithm, the M step does not maximize the objective function but increases it.

This training procedure is similar to the EM algorithm for HMMs, which increases the likelihood of the training data given the correct transcription.

Generative training of HMM models using the EM algorithm, as addressed in Section 2.2.1, is very efficient. On the other hand, generative training for CRFs based on a GEM algorithm is not likely to be as efficient as generative training for HMMs. In this section, links between HMMs and CRFs are established as both of them have an exponential form. This established result will allow any CRF to be constructed from a structurally identical HMM, which was trained by a generative or discriminative process. Hence, CRF initialization may take advantage of the outputs of the well trained HMM models (see Chapter 2).

### 5.2.1 Exponential Family Distributions

HMMs and mixtures of Gaussian models are statistical models belonging to the exponential family (e-family) of distributions. An e-family distribution can always be written in the *canonical* form as shown in equation (5.2.1) [Jebara, 2002].

$$p(\mathbf{o}|\theta) = \exp\big(\mathbf{S}(\mathbf{o})^T\theta - \log \mathbf{Z}(\theta) - \log \mathbf{A}(\mathbf{o})\big) \tag{5.2.1}$$

where $\mathbf{S}(\mathbf{o})$ are the observed sufficient statistics of $\mathbf{o}$, $\theta$ are the natural parameters, and $\mathbf{Z}(\theta)$ is the normalization coefficient (partition function). For example, the Gaussian distribution is characterized by the observed first and second order moments (sufficient statistics) and can be written in the e-family notation (one dimensional case) as

$$\mathbf{S}(\mathbf{o}) = \begin{pmatrix} \mathbf{o}^2 & \mathbf{o} \end{pmatrix} \tag{5.2.2}$$

$$\theta = \begin{pmatrix} -\frac{1}{2\sigma^2} & \frac{\mu}{\sigma^2} \end{pmatrix} \tag{5.2.3}$$

$$\log \mathbf{Z}(\theta) = -\frac{\mu^2}{2\sigma^2} - \log \sigma \tag{5.2.4}$$

$$\log \mathbf{A}(\mathbf{o}) = -\frac{1}{2}\log 2\pi \tag{5.2.5}$$

The mixture of e-family distribution can be written as

$$p(\mathbf{o}|\theta) = \sum_m \alpha_m \exp\left(\mathbf{S}_m(\mathbf{o})^T\theta_m - \log \mathbf{Z}_m(\theta) - \log \mathbf{A}_m(\mathbf{o})\right) \tag{5.2.6}$$

where $m$ represents the incomplete data with $\sum_{m=1}^{M} \alpha_m = 1$ and for $m \in \{1, \dots, M\}$ : $\alpha_m \geq 0$.

### 5.2.2 Exponential Family Activation Functions

CRF models developed in this chapter take advantage of the well established inference and decoding algorithms of the HMM framework. Suppose we define a mixture of e-family (nonlinear) scoring functions or hidden units as

$$b_j(\mathbf{o}|\lambda) = \sum_m \alpha_m \exp\left(\mathbf{S}_m(\mathbf{o})^T\lambda_m - \log \mathbf{Z}_m(\lambda_m) - \log \mathbf{A}_m(\mathbf{o})\right) \tag{5.2.7}$$

where $\lambda$ are unconstrained parameters and $b_j(\mathbf{o}|\lambda)$ and $p(\mathbf{o}|\theta)$ have the same *shape* of parameterization and depend on the same observed sufficient statistics. The only *major* difference between $b(\mathbf{o}|\lambda)$ and $p(\mathbf{o}|\theta)$ is that $b(\mathbf{o}|\lambda)$ is not a density function (i.e. $\int_{\mathbf{o}} b(\mathbf{o}|\lambda)d\mathbf{o} \neq 1$). Note that the mixture components based on e-family activation functions are also not normalized densities. The mixture weights $\alpha_m$ are actually inherited from the mixture of e-family distribution as a mathematical trick to construct the CRF models with minimal modifications to current HMM based systems ($\alpha_m$ are considered to be over-parametrization). This trick will allow us to keep the exact training and decoding algorithms of an HMM implementation. During the training,

$\alpha_m$ are not optimized; the bias term of the scoring function can absorb such offsets and these terms are optimized for each e-family score.

The e-family activation functions of CRF models can be well initialized from similar e-family densities of HMM models. Hence, $b_j(\mathbf{o}|\lambda)$ and $p(\mathbf{o}|\theta)$ will have exact parameters values (i.e. $b_j(\mathbf{o}|\lambda)$ is a density) but after the training procedure the $b_j(\mathbf{o}|\lambda)$ will no longer be considered a density. During this work, we consider e-family scoring functions based on the first and second order characterizing moments. These accumulated sufficient statistics are identical to the e-family Gaussian density accumulated sufficient statistics.

The inference procedure described by Lafferty [Lafferty et al., 2001] is suitable for *single* e-family scoring function based on sufficient statistics of binary or continuous characterizing moments. These two cases are similar to discrete HMMs and single Gaussian HMMs. As with an HMM e-family mixture, we define the probability that the $m^{th}$ score function of the $j^{th}$ state mixture is activated by the observation $\mathbf{o}_t$ as

$$\gamma_{jm}(t) = \gamma_j(t)\frac{\alpha_{jm}b_{jm}(\mathbf{o}_t)}{b_j(\mathbf{o}_t)} \tag{5.2.8}$$

## 5.3 Discriminative Training of CRFs

Despite CRFs being less sensitive to the shape of the data generation than HMMs, neither CRFs or HMMs are exact models of speech generation. Moreover, without infinite training data, Nadas's classical postulates regarding *generative* training are never met [Nadas, 1983].[5] Hence, *discriminative* training will lead to better recognition accuracy. CRF discriminative training may follow an identical setup to HMM

[5] Nadas showed that if the assumptions for prior distribution (i.e. $n$-gram language models) and likelihood distribution (i.e HMM models) are correct, generative training based on MLE and discriminative training based on CML are consistent estimators but CML has a greater variance.

discriminative training. They will differ only in the activation function scoring method and the updating equations, which are defined in this section.

For $R$ training observations $\{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_r, \ldots, \mathbf{O}_R\}$ with corresponding transcriptions $\{w_r\}$, CRF models are trained using the CML criterion, equation (5.3.1), to maximize the posterior probability of the correct word sequence given the acoustic observations.

$$\begin{aligned}\mathcal{F}_{\text{CML}}(\Lambda) &= \sum_{r=1}^{R} \log P_\Lambda(\mathcal{M}_{w_r}|\mathbf{O}_r) \\ &= \sum_{r=1}^{R} \log \frac{P(w_r) \sum_{\mathbf{S}|w_r} \exp \sum_{t=1}^{T} \Big( \sum_i \lambda_{\mathbf{s}_t}^i b_i(\mathbf{o}_t, \mathbf{s}_t, t) + \lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) \Big)}{\sum_{\hat{w}} P(\hat{w}) \sum_{\mathbf{S}|\hat{w}} \exp \sum_{t=1}^{T} \Big( \sum_i \lambda_{\mathbf{s}_t}^i b_i(\mathbf{o}_t, \mathbf{s}_t, t) + \lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) \Big)} \\ &\approx \sum_{r=1}^{R} \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{num}}) - \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{den}})\end{aligned} \tag{5.3.1}$$

which implies a nonparametric CRF formulation similar to an HMM based on equation (5.1.2) and equation (5.1.3). The optimal parameters, $\Lambda^*$, estimated by maximizing the CML criterion, imply minimization of the cross entropy between the correct transcription model and the hypothesized recognition model. In other words, the process maximizes the partition function of the correct models[6] (the numerator term) $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{num}})$, and simultaneously minimizes the partition function of the recognition model (the denominator term) $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{den}})$. The optimal parameters are obtained when the gradient of the CML criterion is zero ( i.e. the two terms of the objective function are equal). The minimization of the denominator partition function will need a full recognition pass (i.e. language model is used in recognition) on all the

[6] Since a summation over potential functions is commonly called the partition function in undirected graphical modelling, we coin the notation $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\text{num}})$ for the summation of all possible state sequences of the correct models.

training utterances for each iteration of this CML training.

### 5.3.1 Numerical Optimization for CRFs

Two hill-climbing methods are introduced to estimate the parameters of the sequential CRFs based on the iterative scaling algorithm [Lafferty et al., 2001]. The two methods will ensure stable update of the objective function, but their speed is a function of the sequence length. Hence, these methods are very slow even for natural language processing tasks. Therefore, sequential CRF models are often trained using gradient based approaches. Many methods are used to accelerate the training process based on the L-BFGS algorithm [Malouf, 2002, Sha and Pereira, 2003], stochastic gradient descent [Gunawardana et al., 2005], and RProp algorithm [Mahajan et al., 2006]. In our work, a variant of gradient ascent optimization [Robinson, 1994] is used to train the models. A review for these training methods is given in Chapter 3. Recently, the CRF training process has been accelerated [Vishwanathan et al., 2006] by using a stochastic meta-descent algorithm [Schraudolph, 2002, Schraudolph, 1999], which utilizes second-order information to adapt the gradient step sizes. Numerical methods are used to train HMMs by relaxing the probabilistic constraints during the HMM training process (the resulting models are CRFs) [Kapadia, 1998]. However, the efficient parameter update of HMMs using the EBW algorithm dominated the discriminative training methods.

For an e-family activation function based on second-order sufficient statistics, the gradient of the CML objective function (see Chapter 3) is given by

$$\nabla\mathcal{F}_{\mathrm{CML}}(1) = \gamma_{jm}^{\mathrm{num}} - \gamma_{jm}^{\mathrm{den}} \tag{5.3.2}$$

$$\nabla\mathcal{F}_{\mathrm{CML}}(\mathbf{O}) = \mathcal{C}_{jm}^{\mathrm{num}}(\mathbf{O}) - \mathcal{C}_{jm}^{\mathrm{den}}(\mathbf{O}) \tag{5.3.3}$$

$$\nabla \mathcal{F}_{\text{CML}}(\mathbf{O}^2) = \mathcal{C}^{\text{num}}_{jm}(\mathbf{O}^2) - \mathcal{C}^{\text{den}}_{jm}(\mathbf{O}^2) \tag{5.3.4}$$

In these equations, $\mathcal{C}_{jm}(1) = \gamma_{jm}$ are the activation function occupancies summed over time. $\mathcal{C}_{jm}(\mathbf{O})$ and $\mathcal{C}_{jm}(\mathbf{O}^2)$ are sums of all observations and all squared observations respectively, weighted by occupancy, for an e-family activation function $m$ of state $j$.

### 5.3.2 Approximate Iterative Scaling for CRFs

A new algorithm to train sequential CRFs based on Approximate Iterative Scaling (AIS) is introduced in this section. The AIS training family was introduced and developed in Chapter 3. This work was motivated by the "No-free-lunch" theorem of optimization, which states that a general purpose universal optimization strategy is impossible to outperform a highly specialized algorithm designed to fit the structure of a specific problem [Ho and Pepyne, 2002]. HMM optimization based on the EBW algorithm is an example of this theory in action. The EBW algorithm, which is a highly specialized algorithm outperformed the general purpose gradient based optimization. Hence, a more efficient training algorithm than gradient based methods may be developed by integrating prior information about the CRF optimization problem. In particular, we integrate prior information from speech recognition based on HMM and MaxEnt optimization based on iterative scaling, which is detailed in Chapter 3.

In a discriminative training setup, the algorithms of the AIS family may update an e-family activation function based on second-order sufficient statistics, as shown in the following equations

$$\lambda_{jm}^{\tau+1}(1) = \lambda_{jm}^{\tau}(1) + \frac{1}{D_{AIS}} \log \frac{\gamma_{jm}^{\text{num}}}{\gamma_{jm}^{\text{den}}} \tag{5.3.5}$$

$$\lambda_{jm}^{\tau+1}(\mathbf{O}) = \lambda_{jm}^{\tau}(\mathbf{O}) + \frac{1}{D_{AIS}} \log \frac{\mathcal{C}_{jm}^{\text{num}}(\mathbf{O})}{\mathcal{C}_{jm}^{\text{den}}(\mathbf{O})} \tag{5.3.6}$$

$$\lambda_{jm}^{\tau+1}(\mathbf{O}^2) = \lambda_{jm}^{\tau}(\mathbf{O}^2) + \frac{1}{D_{AIS}} \log \frac{\mathcal{C}_{jm}^{\text{num}}(\mathbf{O}^2)}{\mathcal{C}_{jm}^{\text{den}}(\mathbf{O}^2)} \tag{5.3.7}$$

where $D_{AIS}$ is called the *learning rate* and $\tau$ is the iteration number.

If the value of the $D_{AIS}$ is sufficiently small, we expect faster training where the CML value will increase at every iteration. However, there is no theoretical guarantee that the algorithm will converge. When $D_{AIS}$ is extremely large, it is easy to prove mathematically that the updates can guarantee an increase in the objective function. The tuning of $D_{AIS}$ is task dependent and the selection of a specific $D_{AIS}$ value, should balance between the WER reduction and the optimization speed, which we measure only in terms of the number of iterations over the training data.

A good heuristic for choosing a suitable $D_{AIS}$ value may be suggested and it is given

$$D_{AIS} = E * \max_{r=1}^{R}(\max_{t=1}^{T_r} \sum_i |b_i(\mathbf{o}_t^r, \mathbf{s}_t, t)|) \tag{5.3.8}$$

where $D_{AIS}$ is actually the maximum value of the summation of the absolute values of acoustic features per frame and $E$ is a global learning rate, which has a default value of $E = 1$ and may vary between 0.5 and 2.0. This heuristic is usually sufficient to ensure the increase of the CML objective function and provide fast training of CRF models or Augmented CRFs (see next Chapter).

There is a rational justification of this heuristic selection and why it works in practical implementations. Sequential CRF models are frame based models and the update equations of the constraints are directly related to sufficient statistics accumulated per frame. If we pretend that the sequential CRF models are simple MaxEnt models by ignoring the state space constraints during the update process, simple maximum frame-wise feature summation may shed light on a suitable value for $D_{AIS}$. Ignoring the state space constraints during the update process, means these parameters are kept fixed during the discriminative training like the language model parameters. Hence, the suggested heuristic does not take into account the update of the transition characterizing moments $a_j(\mathbf{s}_t, \mathbf{s}_{t-1}, t)$. The pragmatic reason behind that is related to an experience in the speech community, which has suggested that updating transition probabilities in HMMs does not lead to improvement in the recognition accuracy.

The described updating equation is suitable for constraints that have positive values. In particular, the constraints related to the bias (occupation) and second order moments. First order moments can have negative values, so the update equation can be similar to the one described in Chapter 3. The disadvantage of this approach is that the first order constraint accumulators will be doubled with respect to the storage requirement of HMMs with similar structure. An alternative solution to this problem is to scale the acoustic features between 0 and 1.

The described AIS algorithm successfully allows us to train CRF models in a practical and more computationally efficient way than gradient based methods given that we run fewer iterations for the estimation of the parameters. This is very important for large vocabulary speech recognition systems that will be built on the top of CRF models. The next section will show some experimental results on the TIMIT database.

## 5.4 Experimental Work

We report results on the TIMIT phone recognition task. We used the 420 speaker training set, analyzed using a 25ms Hamming window at a 10 ms fixed frame rate, resulting in 1,410,069 frames. The baseline 39-feature vector per frame consists of 12 MFCC coefficients, energy, along with the first and second order derivatives. All the acoustic features have been scaled between zero and one. This will maintain the exact storage complexity between HMMs and CRFs during the training process.

The described system is based on context independent left-to-right HMMs. The original 61 phone classes in TIMIT were mapped to a set of 48 labels, which were used for training. The decoding results are reported on a 39 phone classes as described originally in the experiments by Lee [Lee and Hon, 1989]. The whole TIMIT test data is used to report the decoding results. The recognition network constructed based on a bigram language model was estimated from the training data. The language model scaling factor is set to 6.0 during the decoding process.

Table 5.1: *HMM baseline decoding results for TIMIT database*

| #Mix | $PER_{MLE}$ | $Itr_{EBW}$ | $PER_{CML}$ |
|---|---|---|---|
| 1 | 45.7% | 2 | **42.2%** |
| 10 | 30.9% | 3 | **29.2%** |
| 20 | 28.6% | 2 | **27.7%** |
| 40 | 27.2% | 2 | **26.7%** |

The baseline system HMMs have three emitting states and the emission probabilities were modelled with mixtures of Gaussian densities with diagonal covariance matrices. The complexity of the models is increased using a mixture splitting procedure [Young et al., 2001]. The generative HMMs were trained by four iterations with the maximum likelihood criterion. HMMs were refined using CML discriminative training based on the EBW algorithm [Normandin, 1991]. Only the means $\mu_{jm}$ and variances $\sigma_{jm}$ were updated during the CML discriminative training [Burget et al., 2004]. The $D_{EBW}$ is set on the Gaussian level $D_{jm} = \max\{2D^{min}_{jm}, 2\gamma^{den}_{jm}\}$, where $D^{min}_{jm}$ is the value required to ensure positive variances [Woodland and Povey, 2000]. The acoustic model scaling factor is set to 1.0 during the discriminative training of both HMMs and CRFs. The baseline Phone Error Rates (PER) are shown in Table 5.1 and they were reported to the best results on the test set. The actual CML iterations count is reported, since the number of iterations may be important for discriminative training of a speech recognition system.

The CRFs were constructed directly from the discriminatively trained HMMs as described in Section 5.2. Hence, the parameters of the second order moments diagonal activation functions were refined during the training. The training procedure accumulates the $\mathcal{M}^{\text{num}}$ sufficient statistics via a Viterbi pass of the reference transcription and a Baum-Welch pass through the phone recognition network to accumulate the $\mathcal{M}^{\text{den}}$ sufficient statistics. The $D_{AIS}$ constant required for the AIS algorithm is calculated from the training data, which was around 30. Several values of $D_{AIS}$ ($D_{AIS} = 30$, $D_{AIS} = 20$, and $D_{AIS} = 40$) have been tested for the discriminative training of the CRFs based on the AIS algorithm.

To compare the performance of the AIS algorithm with other gradient based algorithms, we train CRFs with the L-BFGS algorithm, Conjugate Gradient (Fletcher-Reeves update) algorithm, and adaptive gradient ascent algorithm, respectively. These CRFs have a single e-family activation function per state (i.e. similar to a single Gaussian per state in HMMs). The initial $\eta$ for the adaptive gradient ascent algorithm is

Table 5.2: *Training methods evaluation for CRFs.*

| Itr | $D_{AIS} = 30$ | $D_{AIS} = 40$ | GRA-adaptive | CG | L-BFGS |
|---|---|---|---|---|---|
| 1 | 41.2% | 41.3% | 44.8% | 41.6% | 41.6% |
| 2 | 41.4% | 41.2% | 41.8% | **41.5**% | 41.5% |
| 3 | 40.9% | 41.0% | 42.3% | 41.9% | 41.9% |
| 4 | 41.1% | 41.0% | 41.6% | 41.6% | **41.5**% |
| 5 | **40.8**% | **40.9**% | **41.4**% | 41.8% | 41.7% |

set to $\frac{1}{D_{AIS}} = \frac{1}{40}$. The first five iterations PER results are shown in Table 5.2. The AIS algorithm yields a quick reduction in the PER from the first iteration. Hence, we did not investigate the gradient based methods for more complex models. Different variants of the AIS algorithm have been investigated. The use of an on-line version of the AIS algorithm leads to a reduction of the number of iterations (less than five iterations) but it did not lead to an improvement of the decoding results. In addition, an adaptive version of the AIS algorithm did not lead to an improvement in the decoding results.

Table 5.3: *TIMIT decoding results for CRFs.*

| #Mix | $D_{AIS}$ | $Itr_{AIS}$ | $PER_{CML}$ |
|---|---|---|---|
| 1 | D=30 | 5 | **40.8**% |
| 10 | D=20 | 2 | **29.0**% |
| 20 | D=20 | 2 | **27.4**% |
| 40 | D=30 | 2 | **26.5**% |

The results of the discriminative training of the CRFs are reported in Table 5.3. For simple CRFs based on a single e-family activation function per state, CRF modelling can lead to a 1.4% reduction in PER over the discriminative HMMs. When the model complexity increases, CRF modelling consistently outperforms discriminatively trained HMMs but the PER reduction is usually less than 1.0% for the TIMIT task. These results may be interpreted as follows: HMMs and CRFs are discrete state space models which are based on the first order Markov assumption. In addition, the two approaches accumulate the same sufficient statistics derived from standard front end processing. As a result, the only difference between results generated by CRFs and HMMs may be due to the form of e-family activation functions and how they are estimated. The e-family activation functions used within the CRF framework are more flexible discriminant functions than the HMM Gaussian activation functions. This may explain the limited success of the more complex CRFs - as compared to HMMs- for the TIMIT corpus. However, for crude models based on a single e-family activation function per state, CRFs show significant improvement with respect to HMMs.[7]

[7] Note that, different training algorithms may also affect the decoding results.

# Chapter 6

# Augmented Conditional Random Fields

Acoustic context information may be incorporated in a speech recognition system as dynamic features [Furui, 1986]. The contextual information is usually implemented by augmenting an acoustic vector with its first and second order derivatives. An alternative approach, used in hybrid ANN/HMM systems, employs a window centered around the current frame with $2c + 1$ frames width ($c$ left frames, current frame, and $c$ of right frames), input to a connectionist phonetic classifier [Morgan and Bourlard, 1995]. We refer to the augmented spaces obtained from such approaches as *low dimensional spaces* as addressed in Chapter 5.

The aim of developing Augmented Conditional Random Fields (ACRFs) is to take advantage of context information in a high dimensional space within the CRF framework. The augmentation process starts by constructing a high dimensional acoustic space with a large number of dimensions ($\approx 10^6$ dimensions). This step is followed by modelling and integrating the acoustic context information into the high dimensional space to model the sequential phenomena of the speech signal. We refer to such spaces as *Augmented Spaces* and they are detailed further in Section 6.1. Augmented Spaces aim to simplify the acoustic classification problem as high dimensional spaces are more likely to be linearly separable than low dimensional spaces. ACRFs are phonetic models that are trained specifically and efficiently to take advantage of Augmented Spaces. In general, Augmented Spaces will lead to a very complex training problem with a very large number of parameters. This chapter will detail how to construct such augmented spaces and propose and introduce the Incremental AIS (IAIS) algorithm designed to train ACRFs with an effective complexity control.

## 6.1 Augmented Spaces

Moving to high dimensional spaces is a key idea to simplify classification problems. This is usually achieved by mapping the low dimensional input space into a high dimensional space, with linear decision boundaries used for classification. As we detailed in Chapter 4, moving to high dimensional spaces can simplify the classification problem as high dimensional or augmented spaces are more likely to be linearly separable than low dimensional spaces [Cover, 1965].

CRF models are dependent on frame based processing. Hence, augmenting the frame vector can simplify the modelling process and decrease the WER. Two augmentation steps are implemented in this work:

- Augmentation based on constructing high dimensional spaces by scoring a large number of optional constraining characterizing functions, or activation functions, for each acoustic vector (i.e. $\mathbf{o}_t \rightarrow \mathbf{o}_t^{\text{Aug}}$). The dimensionality of the constructed space is typically high (60,000 dimensions or more).

- To take advantage of acoustic context, we add the surrounding frames to the current frame to have further augmentation (i.e. the classification of each frame will be a function in $\hat{f}(\mathbf{o}_{t-c}^{\mathrm{Aug}}, \ldots, \mathbf{o}_{t+c}^{\mathrm{Aug}})$ ). The dimensionality of the resultant space will be very high but most of its elements are close to zero and can be pruned.

The two steps of the augmentation process are detailed in Section 6.1.1 and Section 6.1.2.

### 6.1.1 Augmentation By Parametric Constraints

The description of the constraining characterizing functions is an optional implementation issue in which the prior knowledge for different applications is integrated. These constraints or activation functions as described in the previous chapter have characterizing parameters that have been estimated by any means.

In general, the form of the parametric constraints is optional but we are interested in any e-family activation functions or densities since their scores are positive. As a result, the required storage of the AIS accumulators is minimum (see Chapter 3). The limitation of having positive scores calculated by e-family activation will simplify the calculation of the responsibilities or membership scores, which simulate a posterior probability score over a discrete random variable as we will describe later. Samples of these activation functions are:

$$g_i(\mathbf{o}_t; \lambda) = p_i(\mathbf{o}_t \mid \theta) = \mathcal{N}(\mathbf{o}_t; \mu_i, \Sigma_i) \tag{6.1.1}$$

$$g_i(\mathbf{o}_t; \lambda) = \exp\Big( - \frac{\|\mathbf{o}_t - \mu_i\|^2}{2\sigma_i^2} \Big) \tag{6.1.2}$$

$$g_i(\mathbf{o}_t; \lambda) = \exp(\mathbf{o}_t^T \Lambda_i \mathbf{o}_t + \lambda_i^T \mathbf{o}_t + b_{i0}) \tag{6.1.3}$$

$$g_i(\mathbf{o}_t; \lambda) = \exp(\lambda_i^T \mathbf{o}_t + b_{i0}) \tag{6.1.4}$$

The e-family activation functions (6.1.1), (6.1.2), and (6.1.3) can be estimated by accumulating up to second order statistics resulting in *quadratic* discriminant functions. The e-family activation function (6.1.4) is based on *linear* discriminant functions estimated from first order statistics.

In this work, diagonal Gaussian activation functions are used to partition or cluster the acoustic space, which are a flexible model with a strong and rich history in speech recognition. The diagonal Gaussians can be estimated efficiently using the EM algorithm for GMM/HMM models to locate the dense regions in the acoustic space [Dempster et al., 1977]. This is due to the fact that GMMs and continuous density HMMs may be interpreted as vector quantizers and the estimated Gaussians represent the dense regions in the clustering process. As a result, by ignoring all mixture weights or transition matrices in an acoustic model, we have a practical method to locate the dense regions in any acoustic space. The Gaussians can be refined using the Extended Baum-Welch (EBW) algorithm to increase the discrimination of Gaussian distributions [Normandin, 1991]. Hence, the resulting Gaussian models will estimate the likelihood score for an observation, and a normalized version of that likelihood score will take the role of MaxEnt constraints for each state in ACRF framework.

The process of augmenting the low dimensional space to high dimensional space $\mathbf{o}_t \rightarrow \mathbf{o}_t^{\text{Aug}}$, starts with scoring all the activation functions or constraints $i$. Then, the constraints are sorted according to their scores and only the top $n$-best shortlist are selected.[1] Once an $n$-best shortlist is available, the augmented vector is constructed and its size equals the number of constraints in the recognition problem $\mathbb{R}^{\text{Aug}}$. A

[1] Typically, $n$-best nearest-neighbor shortlist size is set to 10.

state constraint value in the new augmented space is calculated as a membership or responsibility score for each parametric constraint and it is given by

$$b_i(\mathbf{o}_t, \mathbf{s}) = \frac{g_i(\mathbf{o}_t; \lambda)}{\sum_i g_i(\mathbf{o}_t; \lambda)} \approx \frac{g_i(\mathbf{o}_t; \lambda)}{\sum_{n\text{best}} g_i(\mathbf{o}_t; \lambda)} \tag{6.1.5}$$

where the normalization step is conceptually redundant. However, its use may be necessary in order to satisfy conditions imposed by the IS algorithm derivation, which uses the Jensen's inequality lower bound in order to decouple all parameters. The use of the Jensen's inequality assumes that the values of the constraints are represented as a posterior probability over a discrete random variable. Consequently normalization, although not theoretically necessary, is useful in order to reduce the number of IS iterations for training.

The augmented vector $\mathbf{o}_t^{\text{Aug}}$ described in (6.1.5) may be similar to the rank based recognition developed in [Likhododev and Gao, 2002]. It was suggested that rank scores may offer a degree of robustness [Hong-Kwang and Gao, 2003]. Given the augmented space concept, the state constraint value in [Likhododev and Gao, 2002] may be defined as a *Rank*[2] for the $n$-best shortlist by

$$b_i(\mathbf{o}_t, \mathbf{s}) = \frac{2}{1+k} \tag{6.1.6}$$

where $k$ is the order of the nearest neighborhood shortlist elements. However, we preferred the responsibility scoring method (6.1.5) for three reasons:

1. The first reason is related to the basic fact that the neighborhood rank is not directly related to the actual calculated scores. To explain this fact, suppose a frame is only activating three activation functions in the shortlist and the

[2] The rank scoring is possibly inherited from $k$ nearest neighbor algorithm, where a rank $\frac{1}{k}$ usually weights the $k$ nearest neighbor rule voting.

size of the $n$-best shortlist is 10, in the rank based augmented spaces, $b_i(\mathbf{o}_t, \mathbf{s})$ for the elements of the shortlist is contributing in the classification process and this may lead to more confusion in the classifications. On the other hand, the membership score in (6.1.5) is more likely to capture this effect by distributing the posterior mass only between the three activation functions.

2. The second reason is related to the constraint formulation of the ACRFs. Since augmented spaces imply large number of parameters, it is desirable to increase the sparsity of the model. The rank based scores will lead to relatively large expectations for unreliable constraints.[3] On the other hand, unreliable constraints based on membership scores will have expectations close to zero. This property will result in sparse models as described in Section 6.3.1.

3. The third reason is also related to the training process as the rank based scoring will lead to slow training.[4] In addition, the training speed is a function of the size of the $n$-best shortlist.

Constructing the augmented high dimensional frame implies scoring all e-family activation functions of the system for each frame to find the index of the $n$-best kernel functions that strongly responsible about activating the acoustic frame (i.e. the most likely acoustic regions that the frame belongs to them). This process is very expensive with exact scoring methods. Fortunately, this problem has been addressed in large vocabulary search and decoding to increase the speed of the recognition process and

[3] The worst constraint in the 10 best rank scoring, will have value $\frac{2}{1+k} = 2/11 \approx 0.18$. This scheme has consistently large scoring values per frame and may lead to large expectations and classification confusion.

[4] Actually, $D_{\text{AIS}} = \sum_{k=1}^{10} \frac{2}{1+k} = 4.0398$ will lead to slow steps in the hill-climbing process. The membership scores will lead to $D_{\text{AIS}} = 1$ and this is the best value to get fast training for ACRF models.

known as Gaussian selection[5] [Bocchieri, 1993, Gales et al., 1999]. Gaussian selection aims to take advantage of the basic fact that if a frame is outside a hyperellipsoid region specified by a Gaussian[6], then the likelihood score of that Gaussian is very small. Hence, only a *nearest neighbor* shortlist (i.e. $n$-best) of Gaussians is selected or activated by that frame.

The Gaussian selection process can be implemented by generating a set of codewords or clusters for all Gaussians in the system. The codewords are obtained with a clustering algorithm of all Gaussians based on similarity measure between two Gaussians. Each codeword points to subset of Gaussians known as the shortest list for that Gaussian. Hence, the likelihood computations for each frame are reduced with a minor sacrifice in the recognition accuracy.

Despite that Gaussian selection methods aim to speed up the recognition process, our motivation for using Gaussian selection methods is to facilitate the construction of the augmented spaces to increase *frame discrimination* within ACRF modelling.

The augmentation process $\mathbf{o}_t \rightarrow \mathbf{o}_t^{\text{Aug}}$ is sketched in Figure 6.1. The augmented space dimensionality equals to number of basis functions. Most of the elements of the augmented vector $\mathbf{o}_t^{\text{Aug}}$ are zero as they are considered outliers for the point $X$. The relative membership scores are obtained for Gaussians near the point $X$. The summation of the elements in the augmented vector is equal to 1.0. Finally, the orientation of the hyperellipsoid axes is parallel to coordinate axes and the $n$-best shortlist is defined by a circle around the point $X$.

Although the augmented vector dimensionality is very high, few elements of the

[5] The idea can be generalized for any activation function given the availability of distance measure between two activation functions of the same functional form.

[6] In a one dimensional Gaussian, this may mean that the point is far from the mean $\mu$ with distance $\gg 2\sigma$.

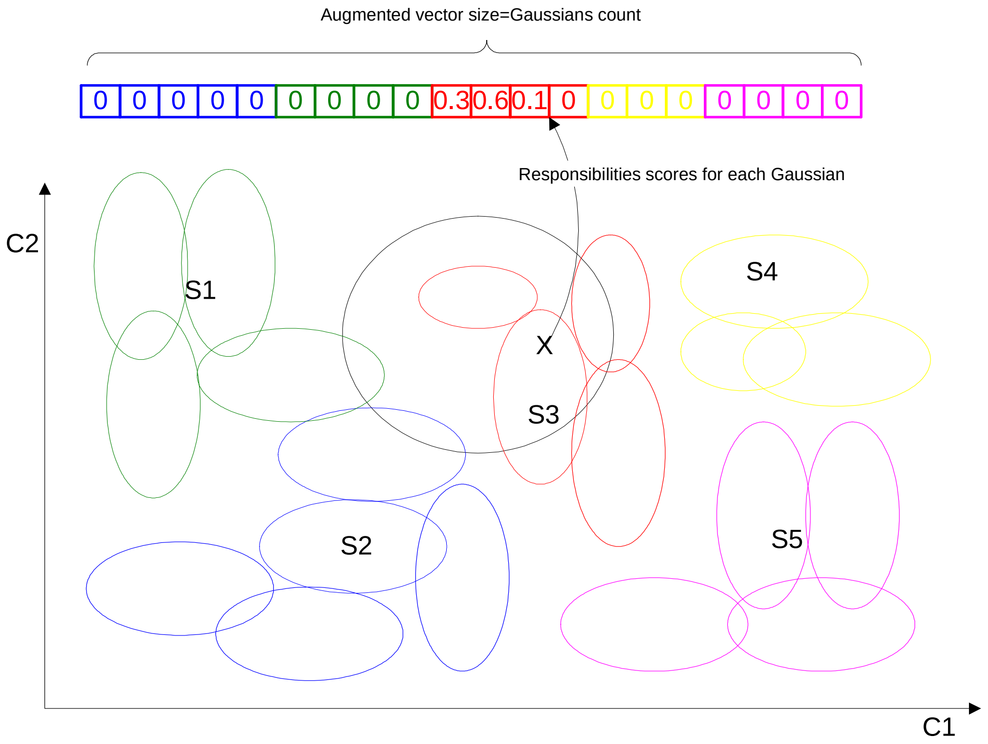

Augmented vector size=Gaussians count
0 0 0 0 0 0 0 0 0 0.3 0.6 0.1 0 0 0 0 0 0 0 0
Responsibilities scores for each Gaussian
C2
S1
S4
X
S3
S2
S5
C1

Figure 6.1: Two dimensional space is partitioned into 20 regions specified by diagonal Gaussians. The augmented space dimensionality is equal to the number of Gaussians (activation functions) in the acoustic space. Hence, the two dimension space is augmented to a twenty dimensional space $\mathbb{R}^2 \rightarrow \mathbb{R}^{20}$. The augmented vector is constructed by calculating a membership score for each Gaussian. The majority of the elements of the augmented vector have very low membership score (i.e. $\approx$ zero). The $n$-best shortlist size $= 3$.

augmented vector are non zero. These few elements are actually represent the effective dimensionality of the augmented vector. For example, if we construct an augmented space with an augmented dimensionality $d^{\text{Aug}} = 2,000,000$ and the $n$-best shortlist size = 10, we refer to effective dimensionality as $d^{\text{Eff}} = 10$.

Intrinsic dimensionality of augmented spaces can be estimated by Karhunen-Loéve transformation or Principle Component Analysis (PCA) [Fukunaga, 1990]. Alternatively, applying Linear Discriminate Analysis (LDA) on the top of the augmented spaces can extract few discriminative coordinates. Note that, PCA and LDA are linear transforms but they can operate in the augmented space. Both augmented PCA and LDA are nonlinear feature extraction methods and they are more practical for large scale problems than kernel PCA and LDA methods [Schlkopf et al., 1998].

### 6.1.2 Augmentation By Adding the Acoustic Context

The augmented vector $\mathbf{o}_t^{\text{Aug}}$ described in the previous section does not take into account acoustic context. Considering acoustic context takes into account a longer time interval for frame discrimination within the modelling process. Longer time intervals have been used in nonparametric discriminant feature transformation/extraction [Hermansky and Sharma, 1998, Hermansky et al., 2000].

Once the high dimensional augmented vector $\mathbf{o}^{\text{Aug}}$ is constructed, it is possible to take advantage of acoustic context by adding the surrounding augmented frames $\mathbf{o}_{t-c}^{\text{Aug}}, \ldots, \mathbf{o}_{t+c}^{\text{Aug}}$ to the current frame during state scoring within ACRF framework. ACRF model has an implicit advantage to model the context, which is inherited directly from the nonparametric formulation of the sequential process. Although dynamic features ( first and second derivative estimates) are commonly used to model acoustic context, they are not used in this work. This is due to that $\Delta$ and $\Delta\Delta$ features are calculated using regression equations, which do not take discrimination between states into account. Hence, adding surrounding context may be more effective in our discriminative training setup. As a result, discriminative regression parameters are estimated during the training process.

Using context modelling will lead to minor change to the computational complexity of augmented space construction but adding these surrounding frames to a sparse augmented vector, will change the problem dimensionality to $(2c+1)d^{\text{Aug}}$ and its effective dimensionality by $(2c+1)d^{\text{Eff}}$. Practically, these augmented spaces complicate the training process and may lead to poor generalization due to the curse of dimensionality [Bellman, 1961]. This problem will be addresses in ACRF optimization in Section 6.3.

## 6.2 ACRF Graphical Model

The aim of developing Augmented Conditional Random Fields (ACRFs) is to take advantage of the context information within the CRF framework. A graphical representation of the dynamic undirected graphical models is shown in Figure 6.2. It is clear that an ACRF model is dependent on arbitrary acoustic observations. This ACRF graphical model is different from the CRF model, which is equivalent to the HMM architecture addressed in Chapter 5.

The conditional distribution behind ACRFs is given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\Big( \sum_{w=t-c}^{t+c} \sum_{i} \lambda^i_{\mathbf{s}_t} b_i(\mathbf{o}_w, \mathbf{s}_t, t) + \lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) \Big) \quad (6.2.1)$$

Equation (6.2.1) combines different state scores (i.e. $\sum_{w=t-c}^{t+c} \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{o}_w, \mathbf{s}_t, t)$) of

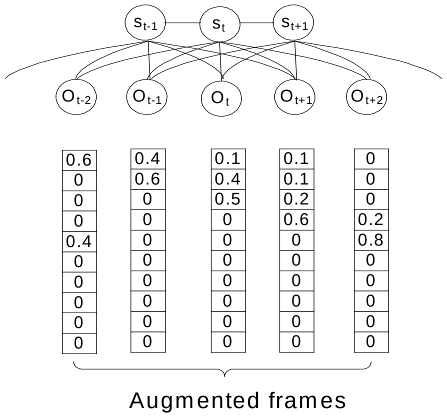


Figure 6.2: ACRF model for phone representation that is dependent on arbitrary acoustic observations.

different streams $w$. The variable $w$ represents the window size of surrounding augmented frames around the current frame. Similarly, if $\mathbf{o}_w$ are calculated from different observation spaces representing an utterance (e.g. MFCC and $f_0$), weighted summation over $w$ can be interpreted as a combination of general stream scores. Hence, stream combination is a natural property within the CRF framework. The state characterizing function $b_i(\mathbf{o}_t, \mathbf{s}_t, t)$ calculation for each frame is addressed in the previous section and $a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) = 1.0$.

Equation (6.2.1) constructs a *linear discriminant* function using the first order statistics accumulated from the augmented space observations. First order statistics may be sufficient to construct linear decision boundaries in high dimensional spaces, since it may be impractical to accumulate the second order statistics during the estimation process to have a full Gaussian functional form. Consequently, the functional form of the constructed state characterizing functions can be understood as an unnormalized spherical Gaussians.

The ACRF model takes advantage of the construction of augmented spaces to model the acoustic context. It may be expected that modelling acoustic context in augmented spaces is a more effective technique in the ACRF framework since the augmented space confusability is less than for low dimensional spaces. This may increase *frame discrimination* within the acoustic modelling process. This is due to the fact that frame based modelling is inadequate to model the speech signal.

## 6.3 ACRF Optimization

ACRF optimization implies the estimation of the Lagrange multipliers of the state characterizing functions associated with augmented spaces and the state transition characterizing functions. In this work, we also take the same pragmatic attitude described in the previous chapter by not updating the Lagrange multipliers related to the state transition characterizing functions. As mentioned before, this will lead to the smallest $D_{\text{AIS}}$ required to update the parameters associated with augmented spaces. Hence, a few iterations of CRF training will usually lead to significant reduction in the WER. This is due to the fact that augmented spaces are *normalized* spaces (i.e. *every* observation in the training corpus has $M(\mathbf{o}) = 1$) and AIS is well-suited to training such spaces as described in Chapter 3. AIS training for a CRF model based on normalized spaces can update all the parameters in the model with an ideal slowing factor $D_{\text{AIS}} = 1.0$. Context modelling will lead to a minor modification to the training process.

ACRF models associated with augmented spaces will lead to a complex training problem with a very large number of parameters. This large number of parameters corresponds to an augmented matrix of size $|\mathbf{s}| \times (2c+1)d^{\text{Aug}}$, where $|\mathbf{s}|$ represents the total number of states in the system and $(2c+1)d^{\text{Aug}}$ dimensionality of the constructed augmented space.[7] Training a large number of parameters will lead to the overfitting phenomenon and poor generalization.

*Regularization* is a common approach to overcome poor generalization problem and to provide effective complexity control. In Chapter 4, an incremental greedy methodology to evaluate the importance of the constraints was utilized and the training stopped based on model comparison within the marginal likelihood framework. However, this method is not efficient for large scale problems. In this work, regularization is achieved by adding an $l_1$ norm penalty term to the CML criterion as described in Section 6.3.1. Based on the $l_1$ regularizer, an efficient incremental algorithm is designed to train ACRF models is described in Section 6.3.2.

### 6.3.1 $l_1$-ACRF Models

Regularization is a common approach to overcome poor generalization and to provide effective complexity control. In this work, regularization is achieved by adding an $l_1$ norm penalty term to the CML criterion:

$$\mathcal{F}_{\text{RCML}}(\Lambda) = \sum_{r=1}^{R} \log Z_{\Lambda}(\mathbf{O}_r|\mathcal{M}^{\text{num}}) - \log Z_{\Lambda}(\mathbf{O}_r|\mathcal{M}^{\text{den}}) - \alpha \sum_{\mathbf{s},d^{Aug}} |\lambda_{ji}| \qquad (6.3.1)$$

where $\lambda_{ji}$ parameter is associated with the $j^{th}$ state and the $i^th$ constraint.

The $l_1$ regularizer[8] or Lasso penalty, $\sum |\lambda_{ji}|$, is often used to increase the sparseness of the model since it can lead to solutions where some elements of $\lambda_{ji}$ are exactly zero [Hastie et al., 2001]. Choosing an $l_1$-norm regularizer will induce sparseness essential for ACRF optimization.

[7] For example, each phone of TIMIT is represented with three states CRFs, this leads to a total number of states equals $48 * 3 = 144$. An augmented space with dimensionality $d^{\text{Aug}} = 129295$, will lead to $18,618,480$ Lagrange multipliers that must be estimated robustly.

[8] [Perkins et al., 2003] details feature selection with $l_0, l_1, l_2$ regularizers for linear models with numerical optimization methods.

As we mentioned in Chapter 4, when the parameters have zero values, selecting or adding those parameters that have the largest absolute value of the gradient to the model $\mathcal{M}$, is a good way to increase the objective function. The gradient of $\mathcal{F}_{\text{RCML}}$ objective function is given by

$$\frac{\partial \mathcal{F}_{\text{RCML}}}{\partial \lambda_{ji}} = \sum_{r=1}^{R} \mathcal{C}_{ji}^{\text{num}}(\mathbf{O}_r|\mathcal{M}^{\text{num}}) - \mathcal{C}_{ji}^{\text{den}}(\mathbf{O}_r|\mathcal{M}^{\text{den}}) - \alpha \text{sign}(\lambda_{ji}) \tag{6.3.2}$$

where the gradient of $\mathcal{F}_{\text{RCML}}$ can be defined for points other than $\lambda_{ji} = 0$ as

$$\frac{\partial |\lambda ji|}{\partial \lambda_{ji}} = \begin{cases} +1 & \text{for } \lambda_{ji} > 0 \\ \text{undefined} & \text{for } \lambda_{ji} = 0 \\ -1 & \text{for } \lambda_{ji} < 0 \end{cases} \tag{6.3.3}$$

Replacing sign$(\lambda_{ji})$ with $-1$ or $+1$ at$\lambda_{ji} = 0$, must be selected to ensure the increase of the $\mathcal{F}_{\text{RCML}}$ objective function and solve the discontinuity of the gradient problem associated with an $l_1$-norm regularizer.

When the gradient of the CML objective function $\frac{\partial \mathcal{F}_{\text{CML}}}{\partial \lambda_{ji}} > \alpha$, this means that $\frac{\partial \mathcal{F}_{\text{RCML}}}{\partial \lambda_{ji}} > 0$ regardless the sign of $\lambda_{ji}$. Since $\lambda_{ji}$ is zero and sign$(\lambda_{ji})$ is not defined, a choice of sign$(\lambda_{ji}) = +1$ can increase the $\mathcal{F}_{\text{RCML}}$ and solves the discontinuity problem of the gradient calculations. Similarly, if $\frac{\partial \mathcal{F}_{\text{CML}}}{\partial \lambda_{ji}} < -\alpha$, this means that $\frac{\partial \mathcal{F}_{\text{RCML}}}{\partial \lambda_{ji}} < 0$ regardless the sign of $\lambda_{ji}$. A choice of sign$(\lambda_{ji}) = -1$ will increase the $\mathcal{F}_{\text{RCML}}$ and solves the discontinuity problem of the gradient calculations. In short, $|\frac{\partial \mathcal{F}_{\text{CML}}}{\partial \lambda_{ji}}| > \alpha$ specifies an *evaluation condition* for useful parameter for modelling in the $l_1$ norm sense. Hence, gradient based optimization can be used to train the models as the gradient is defined and calculated. However, we benefit from the evaluation condition but we train the models using the incremental AIS algorithm.

The parameters where $|\frac{\partial \mathcal{F}_{\text{CML}}}{\partial \lambda_{ji}}| < \alpha$, are not included in the model $\mathcal{M}$ as it is not possible to choose sign$(\lambda_{ji}) = +1$ or sign$(\lambda_{ji}) = -1$ of these parameters, which can lead to an increase in the objective function $\mathcal{F}_{\text{RCML}}$ and solves the discontinuity problem of the gradient calculations. This can explain why an $l_1$ norm will lead to sparse solutions and specify the *maximum number of parameters* can be added to the model $\mathcal{M}$.

The value for the hyperparameter $\alpha$ specifies a compromise between the complexity of the model and modelling accuracy. Increasing the value of $\alpha$ will lead to reduction of the number of the active parameters in the model $\mathcal{M}$. Selecting a suitable value for $\alpha$ can usually be achieved via cross validation. Alternatively, the problem can also be cast as model selection within the marginal likelihood or evidence framework as shown in Chapter 4. A simple and pragmatic method that can be useful for large scale optimization required for speech recognition is proposed. This method is based on training the unregularized ACRFs ($\alpha = 0$) for few iterations and recording the best WER for heldout/test data. Then, the value of $\alpha$ is increased gradually and the $l_1$-ACRFs are retrained. Hence, a good $\alpha$ is selected, which leads to a significant reduction in the number of parameters with minimum reduction in the recognition accuracy. Later, $\alpha$ is fixed during the $l_1$-ACRFs training stage of the context modelling.

Some researchers prune the parameter space by removing the parameters associated with constraints that have low empirical expectation values before the training process[9] [Ratnaparkhi, 1996]. This technique belongs to *learning model parameters* methods only, while fixing the structure of the model. Of course, *learning model structure* is a harder problem with respect to learning the parameters only. Our

[9] Theoretically, we should remove the parameters that do not improve the CML objective function but removing parameters based on low empirical expectations is a practical idea in some cases such as spaces formulated based on binary features.

method learns the model structure and the parameters concurrently in an efficient way.[10]

The discriminative pruning method described here is very practical for large scale problems like speech recognition and will lead to very sparse models as the results show in Section 6.4. Other pruning methods have been developed in neural networks community such as Optimal Brain Damage [LeCun et al., 1990] and Optimal Brain Surgeon [Hassibi et al., 1993]. These methods are based on building large models and using the Hessian matrices of these large models to compute a saliency measure for parameters and make backward model selection. However, building large models and then pruning them is a naive idea for LVCSR systems.

### 6.3.2 Incremental AIS algorithm for $l_1$-ACRFs

The AIS algorithm is an ideal IS algorithm for ACRFs optimization based on normalized spaces since each observation has $M(\mathbf{o}) = \sum_i \mathbf{o}_{ti}^{\text{Aug}} = 1.0$. To show how an AIS algorithm can lead to fast training; the augmented vector $\mathbf{o}_t^{\text{Aug}}$ was designed to have the $\sum_i \mathbf{o}_{ti}^{\text{Aug}} = 1.0$ as shown in Section 6.1. Suppose the number of frames in the longest utterance in the training data is $T$. Since state scores summation per frame $M(\mathbf{o}) = \sum_i \mathbf{o}_{ti}^{\text{Aug}} = 1.0$ and the observation sequence has $T$ frames, the augmented space formulation will lead to a slowing constant $C_s = \sum_{t=1}^{T} \sum_i \mathbf{o}_{ti}^{\text{Aug}} = T$ associated with state constraints. On the other hand, the state transitions are associated with binary features and we have approximately $T$ transitions. This will lead also to a scaling constant $C_t \approx \sum_{t=1}^{T} a(\mathbf{s}_t, \mathbf{s}_{t-1}, t) = T$ associated with transition constraints.

[10] The word *structure* is inspired by Vapnik's Structural Risk Minimization [Vapnik, 1998], where he actually was interested to learn the structure of a model by regularization rather than Empirical Risk Minimization, which learns the parameters of a specified model only.

Hence, the $C$ value to ensure the convergence of CRFs training may be $C_s + C_t = 2T$. For example, if the longest utterance in the training data has 500 frames, $C = 1000$ may guarantee the convergence of the ACRFs training. This large value will lead to very slow training and that is why CRFs are never trained by IS algorithms.

As we described in the previous chapter, the Lagrange multipliers related to the state transition characterizing functions pragmatically are not updated. On the other hand, we use the AIS algorithm with $D_{\text{AIS}} = \sum_i \mathbf{o}_{ti}^{\text{Aug}} = 1.0$ to train ACRFs since CRFs are frame based models and $M(\mathbf{o}_t) = \sum_i \mathbf{o}_{ti}^{\text{Aug}} = 1.0$. Thus, it may be clear that AIS algorithm is very fast with respect to the exact IS algorithm (i.e. for the same example described above, $D_{\text{AIS}} = 1.0$ and $C_{s+t} = 1000$). In the case of context modelling, $D_{\text{AIS}} = w$, where $w$ is the number of frames in the context window $w = 2c + 1$. To keep $D_{\text{AIS}} = w$ is small as possible, it is desirable to make $w = 1$ or $w = 3$ . This may be done using incremental training by adding one or two context frames and train the whole system but only updating the parameters of the new added frames. The process can be repeated by adding new context frames and updating the parameters associated with these frames.

$l_1$-ACRFs were trained using the incremental training algorithm detailed in algorithm 6.1. The algorithm starts with empty model $\mathcal{M} = \phi$, where all the state constraints are not included in the model. At each iteration, the *evaluation condition* $|\frac{\partial \mathcal{F}_{\text{CML}}}{\partial \lambda_{ji}}| > \alpha$ is calculated for each parameter, which is not part of the current model. If any parameter is able to pass this gradient test, we add it to the model $\mathcal{M} \cup \lambda_{ji}(\mathbf{O})$. All the parameters which are elements of the current model $\lambda_{ji}(\mathbf{O}) \in \mathcal{M}$ are updated using AIS step. Finally, the process stops according to a valid termination condition.

**Algorithm 6.1** Incremental AIS algorithm for $l_1$-ACRFs

```
given R training observations $\{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_r, \ldots, \mathbf{O}_R\}$ with corresponding transcriptions $\{w_r\}$.
initialize $\mathcal{M} = \phi$ {empty states (no parameters) in $\mathcal{M}$ }
initialize $D_{\text{AIS}} = 1$ for one augmented frame or $D_{\text{AIS}} = 2c+1$ for context modelling.
initialize $\alpha > 0$ for $l_1$-ACRFs.
initialize $\mathcal{I}$ with the maximum iteration count$\leq 20$.
repeat
  itr = itr + 1
  Accumulate $\mathcal{C}_{ji}^{\text{num}}(\mathbf{O}_r|\mathcal{M}^{\text{num}})$ and $\mathcal{C}_{ji}^{\text{den}}(\mathbf{O}_r|\mathcal{M}^{\text{den}})$, $\forall r$
  for all $\lambda_{ji}(\mathbf{O}) \notin \mathcal{M}$ do
    if $|\mathcal{C}_{ji}^{\text{num}}(\mathbf{O}|\mathcal{M}^{\text{num}}) - \mathcal{C}_{ji}^{\text{den}}(\mathbf{O}|\mathcal{M}^{\text{den}})| > \alpha$ then
      $\mathcal{M} \cup \lambda_{ji}(\mathbf{O})$ {add the parameter to the model $\mathcal{M}$}
    end if
  end for
  for all $\lambda_{ji}(\mathbf{O}) \in \mathcal{M}$ do
    $\lambda_{ji}^{t+1}(\mathbf{O}) = \lambda_{ji}^{t}(\mathbf{O}) + \frac{1}{D_{\text{AIS}}} \log \frac{\mathcal{C}_{ji}^{\text{num}}(\mathbf{O}|\mathcal{M}^{\text{num}})}{\mathcal{C}_{ji}^{\text{den}}(\mathbf{O}|\mathcal{M}^{\text{den}})}$
  end for
  if $D_{\text{AIS}} > 1$ then
    adapt $D_{\text{AIS}}$ {optional process; see Chapter 3 for adaptive AIS}
  end if
until $\mathcal{F}_{\text{CML}}(\Lambda) < \Delta$ or itr $< \mathcal{I}$
```

## 6.4 Experimental Work

We have performed experiments using the TIMIT database. The TIMIT task setup is described in the previous chapter. The acoustic space was partitioned into $\simeq$ 7K e-family (Gaussian) activation functions, which were sufficient to have a high accuracy modelling. The augmented frames are calculated as described in Section 6.1 and they were saved in advance for each utterance in the TIMIT database. Each phone was represented with three states left-to-right ACRF. A special bigram decoder was implemented to decode $l_1$-ACRFs.

All parameters related to state characterizing functions are initialized to zero. The parameters transition scores can be initialized from trained HMM models or from uniform transition matrix. This did not affect the PER results in the experiments but force left to right CRFs. The training procedure accumulated the $\mathcal{M}^{\text{num}}$ sufficient statistics via a Viterbi pass of the reference transcription using HMMs trained using MLE and, Baum-Welch (BW) pass through the all possible phones to accumulate the $\mathcal{M}^{\text{den}}$ sufficient statistics as described in the previous chapter. The $\mathcal{M}^{\text{den}}$ sufficient statistics are approximated with state level $\mathcal{M}^{\text{den}}$ to avoid building lattices in the context modelling stage [Hifny et al., 2005]. The models were trained using the Incremental AIS algorithm described in Section 6.3.2. Incremental AIS can usually train $l_1$-ACRFs without any acoustic context in few iterations.

Since there is no direct relationship between PER and the CML objective function, we report the decoding results in Table 6.1. Since all parameters of $l_1$-ACRF models were initialized from zero, the reported results may suggest that the training process is extremely fast, where the effective training occurred in the first two iterations. This fast training speed is due to the fact that augmented spaces are normalized spaces and the features of the augmented spaces are approximately uncorrelated (i.e. the relation between features is a soft Winner-Take-All relationship).[11] Hence, using quadratic approximations to accelerate the training process may not be useful to train $l_1$-ACRFs. In addition, $l_1$-ACRFs incremental training with a few number of iterations as we suggest may force the L-BFGS algorithm to behave like gradient methods since it may be difficult to have a good estimate of the approximate Hessian matrix that may accelerate the training process.

Table 6.1: *$l_1$-ACRFs decoding results where the hyperparameter $\alpha = 0$.*

| #Itr | Corr | Sub | Del | Ins | PER |
|---|---|---|---|---|---|
| 1 | 73.4 | 18.3 | 8.3 | 3.3 | 29.9% |
| 2 | 74.9 | 17.7 | 7.4 | 3.7 | 28.8% |
| 3 | 75.3 | 17.5 | 7.1 | 3.9 | 28.5% |
| 4 | 75.5 | 17.4 | 7.0 | 3.9 | 28.4% |
| 5 | 75.7 | 17.4 | 6.9 | 4.0 | **28.3**% |

To evaluate the efficiency of the Incremental AIS algorithm for $l_1$-ACRFs, the decoding accuracy for different values of the hyper parameter $\alpha$ is given in Table 6.2. We define our own objective criteria to measure the sparseness of the models. The criteria is measured by the Compression Ratio (CR) of the parameter space and is given by

$$\mathrm{CR}(\alpha) = \frac{\#\mathrm{Param}(\alpha = 0.0) - \#\mathrm{Param}(\alpha)}{\#\mathrm{Param}(\alpha = 0.0)} \tag{6.4.1}$$

The results show how much the incremental training is effective to train $l_1$-ACRFs. The compression ratio with respect to the unregularized model was a practical way to define $\alpha$ as we described in Section 6.3.1. It is clear that increasing the value of $\alpha$, increases the sparseness of the model. It can be seen that for $\alpha = 20$, the active

[11] Context modelling may lead to correlation between some features in the augmented spaces.

parameters were compressed to CR=96% with only 2% absolute or 7.4% relative reduction in the PER. Since a frame can activate only a specific number of regions or partitions in the acoustic space (i.e. regions related to the confusable sounds produced by the vocal tract system), high compression ratios should be expected with a limited loss in the recognition accuracy. The hyperparameter $\alpha = 2$ was chosen for context modelling for next steps and it effectively compresses the parameters with 80%. The language model scaling factor (lms) was fixed to 2.0.

Table 6.2: *$l_1$-ACRFs trained with different hyperparameters $\alpha$. All ACRF models were trained by five iterations only.*

| $\alpha$ | #Param($\alpha$) | CR($\alpha$) | Corr | Sub | Del | Ins | PER |
|---|---|---|---|---|---|---|---|
| 0.0 | 337927 | 0 | 75.7 | 17.4 | 6.9 | 4.0 | 28.3% |
| 0.01 | 337926 | $\approx 0$ | 75.7 | 17.4 | 6.9 | 4.0 | 28.3% |
| 0.1 | 337023 | $\approx 0$ | 75.7 | 17.4 | 6.9 | 4.0 | 28.3% |
| 0.5 | 292592 | 0.13 | 75.6 | 17.4 | 6.9 | 4.0 | 28.4% |
| 1.0 | 179428 | 0.47 | 75.5 | 17.5 | 7.0 | 4.0 | 28.5% |
| 2.0 | **70109** | 0.79 | 75.4 | 17.5 | 7.1 | 4.0 | **28.6**% |
| 5.0 | 33659 | 0.90 | 75.3 | 17.8 | 6.8 | 4.4 | 29.1% |
| 10.0 | 20385 | 0.94 | 75.2 | 18.1 | 6.7 | 4.9 | 29.7% |
| 20.0 | **11990** | 0.96 | 75.2 | 18.2 | 6.6 | 5.6 | **30.4**% |

We further augment the augmented vector $\mathbf{o}_t^{\text{Aug}}$ to $\mathbf{o}_{t-c}^{\text{Aug}}, \ldots, \mathbf{o}_{t+c}^{\text{Aug}}$ by adding context to take into account longer time interval for frame discrimination. The basic decoding results are summarized in Table 6.3. For $c = 1$ and $c = 3$, the lms scaling factor was set to 2.0. For $c > 3$, the lms was set to 1.0. Clearly, it can be seen that considering long acoustic context can lead to significant improvement in PER over the baseline system $c = 0$ (similar to HMMs). These results may suggest that acoustic context is very effective technique to improve the PER.

Table 6.3: *Context modelling within the $l_1$-ACRF framework.*

| $c$ | $d^{\text{Aug}}$ | $d^{\text{Eff}}$ | #Param($\alpha = 2.0$) | Corr | Sub | Del | Ins | PER |
|---|---|---|---|---|---|---|---|---|
| 0 | 6805 | 10 | 70109 | 75.4 | 17.5 | 7.1 | 4.0 | 28.6% |
| 1 | 20415 | 30 | 212447 | 76.8 | 16.6 | 6.6 | 4.0 | 27.1% |
| 3 | 47635 | 70 | 542540 | 78.3 | 15.2 | 6.5 | 3.2 | 24.9% |
| 5 | 74855 | 110 | 937668 | 79.2 | 14.4 | 6.3 | 3.3 | 24.0% |
| 7 | 102075 | 150 | 1388027 | 79.2 | 13.9 | 7.0 | 2.7 | 23.5% |
| 9 | 129295 | 190 | 1875519 | 79.1 | 13.5 | 7.4 | 2.3 | **23.2**% |

## 6.5 $l_1$-ACRFs Evaluation

In this thesis, we were interested in discrete state spaces models like HMMs and their nonparametric twin CRFs and we limit our evaluation discussion to these models only. In general, the evaluation of different acoustic modelling methods based on the WER criterion only may not be accurate and objective. There are many factors that are important, which may not be reported to have accurate evaluation of any system. These factors are related to the speed of training algorithms, number of parameters, how the hyperparameters are tuned, scalability for large vocabulary systems, and how much actually an algorithm is reinventing the wheel. However, Table 6.5 summaries some results on TIMIT recognition task[12], where $l_1$-ACRF models outperform the most accurate published systems [Young and Woodland, 1994, Robinson, 1994, Halberstadt and Glass, 1998, Omar and Hasegawa-Johnson, 2003, Schwarz et al., 2004].

Speech recognition problems involves sequential pattern classification and the HMMs or CRFs are a natural choice to warp the time axis and model the temporal phenomena in the speech signal. It turns out that there are three factors that

[12] The best TIMIT classification results (18.3%) were reported in [Halberstadt and Glass, 1998], where the classification process relies on the correct segmentation provided by a manual segmentation.

Table 6.4: *Comparison between different approaches for TIMIT phone recognition task*

| paper | method | PER |
|---|---|---|
| Young94 | Tied state tri-phone HMMs | 27.7% |
| Robinson94 | Recurrent NN (RNN) | 25.0% |
| Schwarz04 | Nonlinear TRAP transformation | 24.5% |
| Halberstadt98 | Heterogeneous measurement | 24.4% |
| Omar03 | Nonlinear ICA transformation | 24.4% |
| Hifny06 | monophone $l_1$-ACRFs | **23.2**% |

may lead to significant improvements in the recognition accuracy and they address the spectral variability problem. Two of them are related to pattern classification and one is related to sequential processing:

- Augmenting the *state space* can significantly increase the recognition accuracy as described in Chapter 2. In our work, moving from one state CRFs to three states CRFs lead to significant improvements.

- Augmenting the *observation space* may lead to better accuracy by simplifying the classification problem. Although this idea was not common in HMM based recognition systems, moving to high dimensional spaces was the core idea in nonparametric classifiers.

- Decoding the speech signal based on longer time intervals may improve the recognition accuracy. This may be done by integrating acoustic context information in the modelling process to take into account interframe correlation. Adding the first and second order derivatives of the basic acoustic vector was the choice of HMM based systems to integrate such knowledge. Within the

HMM framework, discriminative feature projection is a major tool to implicity integrate the acoustic context information [Hermansky and Sharma, 1998, Hermansky et al., 2000, Povey et al., 2005, Povey, 2005].

$l_1$-ACRF acoustic models take advantage of all these ideas in the same framework. $l_1$-ACRFs can inherit the augmented state space directly from the HMMs. We can even initialize tied state tri-phone $l_1$-ACRFs from any tied state tri-phone HMMs system. $l_1$-ACRFs are developed to take advantage of augmented observation spaces directly and we show how to train them efficiently in few iterations. $l_1$-ACRFs are able to take advantage of acoustic context information in low dimensional representations like HMMs and in the augmented observation spaces, which can lead to significant improvement in PER.

To have a fair comparison to our work, we compare our work with the fMPE framework, which is the state-of-the-art speech recognition systems [Povey, 2005, Povey et al., 2005]. The fMPE framework is a nonlinear discriminant feature projection method, which was developed within the HMM framework. Consequently, it is a front end feature extraction method, which can be useful within the HMMs and $l_1$-ACRFs. The fMPE framework estimates an augmented projection matrix $M$, which is used to construct a new feature vector (i.e. $\mathbf{o}_t^{\text{new}} = \mathbf{o}_t + M\mathbf{o}_t^{\text{Aug}}$). As a result, fMPE estimates a correction factor based on a projection method to improve the discrimination between speech classes and it does not change the system's dimensionality (i.e. $\mathbf{o}_t^{\text{new}}$ and $\mathbf{o}_t$ have the same dimensionality). This system is based on augmented observation spaces and it is trained using a discriminative training method based on the MPE criterion. In an improved fMPE formulation [Povey, 2005], the original set of Gaussians in an acoustic model (which may number in the millions) is re-clustered into a smaller set of Gaussians which simplifies the construction of an augmented space (i.e. its dimensionality will be approximately of order $10^4$). In general, feature projection will lead to undesirable information loss. The $l_1$-ACRFs are directly formulated directly on the top of the augmented spaces and thus eliminate information loss due to any feature projection. Furthermore, the $l_1$-ACRF framework incorporates native decoding algorithms -these being essentially HMM decoders with minor variations - which process information contained in these augmented spaces. Moreover, the $l_1$-ACRF framework integrates a scalable discriminant compression algorithm to prune the redundant parameters. Finally, the $l_1$-ACRF framework parameter optimization method is based on the AIS algorithm or one of its variants. The fMPE framework uses a batch gradient descent algorithm for optimizing the augmented matrix parameters.

## 6.6 Summary

In this chapter, the $l_1$-ACRF framework has been introduced and developed to address the acoustic modelling problem. The main purpose of developing these models is to integrate the acoustic context information in a data driven, sparse, augmented space. To improve discrimination, the acoustic modelling problem was reformulated in high dimensional spaces to increase the discrimination between confusable states. Although kernel spaces are theoretically attractive high dimensional spaces, they are not efficient for frame based acoustic modelling (see Chapter 4). Consequently, augmented spaces, which are computationally efficient high dimensional spaces, were proposed and developed as an alternative to kernel spaces. The acoustic context information has been integrating explicitly into the augmented spaces to increase the discrimination within the acoustic modelling process. Integrating the acoustic context information in high dimensional spaces leads to a dramatic increase of the number of the parameters and a complex training process. Hence, a scalable discriminant compression algorithm to integrate the acoustic context was proposed and developed. The test results have shown significant improvements over the conventional HMM systems. The main difference between the $l_1$-ACRF framework and the HMM framework is that the $l_1$-ACRF framework can explicitly handle the sequential phenomena of the speech signal in an augmented space. Within the $l_1$-ACRF framework, efficient implementation and optimization techniques were developed to ensure the scalability to LVCSR systems.

# Chapter 7

# Conclusions

In this thesis, a new acoustic modelling paradigm has been developed and investigated. The main goal was to improve the discrimination between speech classes by formulating the acoustic modelling problem in high dimensional spaces and explicitly integrating acoustic context information, which conceptually can improve speech recognition system beyond the conventional HMM framework. These ideas were developed within a flexible CRF framework to estimate conditional models used to decode the speech signal.

## 7.1 Summary

A flexible statistical framework for acoustic modelling based on conditional random fields was utilized to investigate approaches to improve the speech recognition accuracy. Conditional random fields are undirected graphical models developed in the context of the MaxEnt principle. They are the most unbiased distributions (direct models) to model the given data, where the shape of the stochastic process generation is not imposed. A major property of CRFs is that the prior information about a problem is integrated via constraints or potential functions representing the sufficient statistics observed from a stochastic process. These constraints can be correlated or statistically independent and are an optional design issue, leading to a flexible statistical framework. Hence, CRFs are theoretically rigorous models - the HMM framework is not a mathematically elegant approach as it applies to the use of dynamic features - to relax the conditional independence properties, which may be considered a basic limitation of the HMM framework.

Discriminative training for CRFs have been addressed in this work. A family of training algorithms using approximate iterative scaling has been introduced and developed to accelerate the training process. These methods are used to train all models developed in this work including the most successful acoustic model paradigm based on the $l_1$-ACRF framework. A main finding related to the training algorithms was that the training speed of the algorithms is a function of the relation between the constraints (i.e. features) in the constructed spaces. For example, $l_1$-ACRF training was efficient since these models are based on normalized spaces and the constraints are approximately independent. In addition, approximate iterative scaling methods lead to fast training for CRFs based on kernel spaces and CRFs based on exponential activation functions. In general, $l_1$-ACRFs can be trained in few batch iterations, like HMMs so a key idea behind the success of HMMs is kept. $l_1$-ACRF acoustic models can be initialized directly from an analogous HMM-based system, which allows them to be readily implemented in most modern recognition systems.

To improve discrimination, the acoustic modelling problem was reformulated in high dimensional spaces to increase the discrimination between confusable states. Three methods of constructing high dimensional high dimensional spaces were investigated. Firstly, feature spaces based on kernel methods, which have dimensionality related to the number of frames. In a classification task, kernel machines based on MaxEnt models showed some success. However, frame based acoustic modelling may limit the scalability of these spaces to develop acoustic models based on the whole training data. Hence, these spaces may be useful for small scale problems or used for pre/post processing. Secondly, feature spaces constructed by adjusting the weights of e-family activation functions within the sequential CRF framework. The e-family activation functions were trained using a variant of the approximate iterative scaling and were initialized from Gaussian activation functions trained by EM or EBW algorithms. The construction of these spaces is very similar to the implicit high dimensional spaces constructed by the Gaussian distributions within the HMM framework. The decoding results show some success compared with HMM decoding results. Although the e-family activation functions are flexible discriminant functions, the limited success with respect to the Gaussian activation functions used within the HMM framework may be related to the fact that the two systems are dependent on the same second order sufficient statistics. Finally, augmented spaces, which are computationally efficient high dimensional spaces, were proposed. Augmented spaces are constructed using vector quantization techniques and they aim to locate the dense regions in the acoustic space with an arbitrary resolution set in advance. Augmented spaces are very practical because they can be constructed from any HMM based acoustic model developed for a certain task.

Short time signal analysis or frame based modelling is a strong limitation to modelling the speech signal for speech recognition. Explicit acoustic context modelling may increase discrimination within the acoustic modelling process since computing state scores over longer time intervals may reduce the acoustic confusability (analogous to phone spectral properties being more clear over longer time intervals in human spectrogram reading). $l_1$-ACRF models take advantage of the construction of augmented spaces to model the acoustic context explicitly in the augmented spaces. Explicit acoustic context modelling may be a more effective technique in augmented spaces rather than low dimensional spaces since confusability in augmented spaces is less than in low dimensional spaces. Integrating the acoustic context into the augmented spaces gave 5.4% absolute improvement with respect to CRFs without any acoustic context (similar to HMMs) as shown in Chapter 6. The results in Table 6.3 may justify why acoustic context can help to improve discrimination. These results show that most improvements in the PER are due to a reduction in substitution errors. This result may suggest that using acoustic context information to compute acoustic scores may be understood as smoothing the state scores as trajectories over longer time intervals. Hence, acoustic context may prevent abrupt jumps in the state space due to short time signal analysis limitations and strong confusability at frame level between similar speech classes. The acoustic context modelling aims to penalize abrupt movements in the state space by integrating prior information collected from the surrounding frames. Integrating the acoustic context information in high dimensional spaces leads to a dramatic increase of the number of the parameters and a complex training process. Hence, a scalable discriminant compression algorithm to integrate the acoustic context was introduced and it is the key idea behind $l_1$-ACRF acoustic modelling. This compression algorithm offers full control, allowing pruning of the parameter space without any computational cost during the training process.

Frame based acoustic models based on $l_1$-ACRFs and HMMs have some similarities; in particular, both approaches have similar training speed and decoding algorithms. Hence, $l_1$-ACRF acoustic modelling attempts to address some of the weak aspects of HMMs while maintaining many of the good aspects, which have made them successful. Within the $l_1$-ACRF framework, the use of high dimensional spaces to reduce confusability and the use of the acoustic context information to handle the sequential phenomena of the speech signal lead to sparse context modelling in an augmented space, which fundamentally can improve speech recognition.

## 7.2 Future Work

While the basic mathematical theory and an efficient implementation of $l_1$-ACRF acoustic modelling have been achieved, there are some aspects, which were not addressed in this work.

It is desirable to test this approach for large vocabulary speech recognition tasks such as Switchboard. Within a large task, many issues will need to be further examined and developed. Firstly, it may be expected that by augmenting the state space using tied-state tri-phones or quin-phones, $l_1$-ACRFs may improve the modelling accuracy as occurs in HMM based acoustic modelling. Rather than developing new clustering algorithms to perform the augmentation process, it may be done efficiently by initializing $l_1$-ACRF system from a similar HMM system. Secondly, the scalability of the discriminative compression algorithm to integrate the acoustic context will need further investigation to ensure high compression ratios given a high modelling accuracy. In addition, the actual gain from the sparse context modelling in a high dimensional space may be correlated with the state space augmentation process.

Thirdly, training $l_1$-ACRFs based on a discriminative criterion such as MPE, which directly minimizes the expected word or phone error rates may be investigated. Finally, speaker adaptation may need further research within the $l_1$-ACRF framework.

# Bibliography

[Afify, 2005] Afify, M. (2005). Extended Baum-Welch reestimation of Gaussian mixture models based on reverse Jensen inequality. In *Proc. INTERSPEECH*, pages 1113–1116, Lisbon, Portugal.

[Aizerman et al., 1964] Aizerman, M. A., Braverman, E. M., and Rozonoer, L. I. (1964). The probability problem of pattern recognition learning and the method of potential functions. *Automation and Remote Control*, 25:1307–1323.

[Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

[Allwein et al., 2001] Allwein, E. L., Schapire, R. E., and Singer, Y. (2001). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141.

[Aronszajin, 1950] Aronszajin, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

[Bahl et al., 1991] Bahl, L., deSouza, P., Gopalakrishnan, P., Nahamoo, D., and Picheny, M. (1991). Decision trees for phonological rules in continuous speech. In *Proc. IEEE ICASSP*, volume 1, pages 185– 188, Toronto, Canada.

[Bahl et al., 1986] Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. IEEE ICASSP*, pages 49–52, Tokyo, Japan.

[Bellman, 1961] Bellman, R. (1961). *Adaptive Control Processes.* Princeton University Press.

[Bentley, 1975] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

[Berger, 1997] Berger, A. (1997). The improved iterative scaling algorithm: A gentle introduction. www.cs.cmu.edu/afs/ aberger/www/ps/scaling.ps.

[Berger, 1985] Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis.* Springer, 2nd edition.

[Beyerlein, 1998] Beyerlein, P. (1998). Discriminative model combination. In *Proc. IEEE ICASSP*, volume 1, pages 481–484, Seattle, Washington, USA.

[Bilmes, 1998] Bilmes, J. (1998). Data-driven extensions to HMM statistical dependencies. In *Proc. ICSLP*, pages 69–72, Sydney, Australia.

[Bilmes, 1999] Bilmes, J. (1999). Buried Markov models for speech recognition. In *Proc. IEEE ICASSP*, volume 2, pages 713–716, Phoenix, Arizona.

[Bilmes, 2006] Bilmes, J. (2006). What HMMs can do. *IEICE Transactions on Information and Systems*, E89-D(3):869–891.

[Bilmes and Bartels, 2005] Bilmes, J. and Bartels, C. (2005). Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, 22(5):89–100.

[Bilmes and Zweig, 2002] Bilmes, J. and Zweig, G. (2002). The graphical models toolkit: An open source software system for speech and time-series processing.

[Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

[Bocchieri, 1993] Bocchieri, E. (1993). Vector quantization for the efficient computation of continuous density likelihoods. In *Proc. IEEE ICASSP*, volume 2, pages 692–694, Minneapolis, MN, USA.

[Boser et al., 1992] Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proc. of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152.

[Bourlard and Dupont, 1996] Bourlard, H. and Dupont, S. (1996). A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proc. ICSLP*, volume 1, pages 426–429, Philadelphia, PA.

[Bridle, 1990a] Bridle, J. (1990a). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Fogelman-Soulie, F. and Herault, J., editors, *Neurocomputing - Algorithms, Architectures and Applications*, pages 227–236. Springer-Verlag.

[Bridle, 1990b] Bridle, J. (1990b). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Proc. NIPS*, volume 2, pages 211–217.

[Brown, 1987] Brown, P. F. (1987). *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Burget et al., 2004] Burget, L., Schwarz, P., Karafiat, M., and Cernocky, H. (2004). *HMM Toolkit STK from Speech@FIT.*

[Cerisara et al., 1999] Cerisara, C., Haton, J.-P., and Fohr, D. (1999). Towards a global optimization scheme for multi-band speech recognition. In *Proc. EUROSPEECH*, pages 587–590, Budapest, Hungary.

[Chang and Lin, 2001] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.* Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[Chen and Rosenfeld, 1999] Chen, S. F. and Rosenfeld, R. (1999). A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University.

[Chou et al., 1992] Chou, W., Juang, B. H., and Lee, C. H. (1992). Segmental GPD training of HMM based speech recognizer. In *Proc. IEEE ICASSP*, volume 1, pages 473–476, San Francisco, CA, USA.

[Chou et al., 1993] Chou, W., Lee, C. H., and Juang, B. (1993). Minimum error rate training based on N-best string models. In *Proc. IEEE ICASSP*, volume 2, pages 652–655, Minneapolis, MN, USA.

[Chou and Reichl, 1999] Chou, W. and Reichl, W. (1999). Decision tree state tying based on penalized Bayesian information criterion. In *Proc. IEEE ICASSP*, volume 1, pages 345–348, Phoenix, Arizona.

[Chow, 1990] Chow, Y.-L. (1990). Maximum Mutual Information estimation of HMM parameters for continuous speech recognition using the N-best algorithm. In *Proc. IEEE ICASSP*, pages 701–704, Albuquerque, NM.

[Clarkson and Moreno, 1999] Clarkson, P. and Moreno, P. J. (1999). On the use of support vector machines for phonetic classification. In *Proc. IEEE ICASSP*, volume 2, pages 585–588, Phoenix, Arizona.

[Collins et al., 2000] Collins, M., Schapire, R. E., and Singer, Y. (2000). Logistic regression, AdaBoost and Bregman distances. In *Proc. of the Thirteenth Annual Conference on Computational Learning Theory*, pages 158–169.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[Cover and Thomas, 1991] Cover, T. and Thomas, J. (1991). *Elements of Information Theory.* Wiley.

[Cover, 1965] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334.

[Darroch and Ratcliff, 1972] Darroch, J. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

[Della Pietra et al., 1997] Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

[Deng et al., 1994] Deng, L., Aksmanovic, M., Sun, X., and Wu, J. (1994). Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. *IEEE Transactions on Speech and Audio Processing*, 2(4):507–520.

[Deterding, 1989] Deterding, D. (1989). *Speaker Normalisation for Automatic Speech Recognition*. PhD thesis, University of Cambridge.

[Digalakis, 1992] Digalakis, V. V. (1992). *Segment-based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*. PhD thesis, Boston University.

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. J. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience, 2nd edition.

[Ellis and Bilmes, 2000] Ellis, D. and Bilmes, J. (2000). Using mutual information to design feature combinations. In *Proc. ICSLP*, volume 3, pages 79–82, Beijing, China.

[Ellis, 2000] Ellis, D. P. W. (2000). Stream combination before and/or after the acoustic model. In *Proc. IEEE ICASSP*, volume 3, pages 1635–1638, Istanbul, Turkey.

[Fahlman, 1988] Fahlman, S. E. (1988). An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, Carnegie Mellon University.

[Fiscus, 1997] Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recogniser Output Voting Error Reduction (ROVER). In *Proc IEEE ASRU Workshop*, pages 347–352, Santa Barbara, CA.

[Frankel and King, 2005] Frankel, J. and King, S. (2005). A hybrid ANN/DBN approach to articulatory feature recognition. In *Proc. INTERSPEECH*, Lisbon, Portugal.

[Frankel and King, 2007] Frankel, J. and King, S. (2007). Speech recognition using linear dynamic models. *IEEE Transactions on Speech and Audio Processing*, In press.

[Fukunaga, 1990] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition.

[Furui, 1986] Furui, S. (1986). Speaker independent isolated word recognizer using dynamic features of the speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59.

[Gales, 1999a] Gales, M. (1999a). Maximum likelihood multiple projection schemes for hidden Markov models. Technical Report CUED/F-INFENG/TR365, Cambridge University.

[Gales, 1999b] Gales, M. (1999b). Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions Speech and Audio Processing*, 7(3):272–281.

[Gales et al., 1999] Gales, M., Knill, K., and Young, S. (1999). State-based Gaussian selection in large vocabulary continuous speech recognition using HMMs. *IEEE Transactions on Speech and Audio Processing*, 7(2):152–161.

[Gales and Young, 1993] Gales, M. and Young, S. (1993). The theory of segmental hidden Markov models. Technical Report CUED/F-INFENG/TR.133, Cambridge University.

[Gallager, 1968] Gallager, R. G. (1968). *Information Theory and Reliable Communication*. Wiley.

[Gao, 2003] Gao, Y. (2003). Coupling vs. unifying: Modeling techniques for speech-to-speech translation. In *Proc. EUROSPEECH*, pages 365 – 368, Geneva, Switzerland.

[Gold and Morgan, 1999] Gold, B. and Morgan, N. (1999). *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Wiley.

[Gopalakrishnan et al., 1991] Gopalakrishnan, P. S., Kanevsky, D., Nadas, A., and Nahamoo, D. (1991). An inequality for rational function with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, 37(1):107–113.

[Guiasu and Shenitzer, 1985] Guiasu, S. and Shenitzer, A. (1985). The principle of maximum entropy. *The Mathematical Intelligencer*, 7(1):42–48.

[Gunawardana and Byrne, 2001] Gunawardana, A. and Byrne, W. (2001). Discriminative speaker adaptation with conditional maximum likelihood linear regression. In *Proc. EUROSPEECH*, pages 1203–1206, Aalborg, Denmark.

[Gunawardana et al., 2005] Gunawardana, A., Mahajan, M., Acero, A., and Platt, J. (2005). Hidden conditional random fields for phone classification. In *Proc. INTERSPEECH*, pages 1117–1120, Lisbon, Portugal.

[Haeb-Umbach and Ney, 1992] Haeb-Umbach, R. and Ney, H. (1992). Linear discriminant analysis for improved large vocabulary speech recognition. In *Proc. IEEE ICASSP*, volume 1, pages 13–16, San Francisco, CA, USA.

[Hagen and Bourlard, 2001] Hagen, A. and Bourlard, H. (2001). Error correcting posterior combination for robust multi-band speech recognition. In *Proc. EUROSPEECH*, pages 257–260, Aalborg, Denmark.

[Hagen et al., 1998] Hagen, A., Morris, A., and Bourlard, H. (1998). Subband-based speech recognition in noisy conditions: The full combination approach. Technical Report IDIAP-RR98-15, IDIAP.

[Halberstadt and Glass, 1998] Halberstadt, A. and Glass, J. (1998). Heterogeneous measurements and multiple classifiers for speech recognition. In *Proc. ICSLP*, volume 3, pages 995–998, Sydney, Australia.

[Hasegawa-Johnson et al., 2005] Hasegawa-Johnson, M., Baker, J., Borys, S., Chen, K., Coogan, E., Greenberg, S., Juneja, A., Kirchhoff, K., Livescu, K., Mohan, S., Muller, J., Sonmez, K., and Wang, T. (2005). Landmark-based speech recognition: Report of the 2004 Johns Hopkins summer workshop. In *Proc. IEEE ICASSP*, volume 1, pages 213– 216, Philadelphia.

[Hassibi et al., 1993] Hassibi, B., Stork, D. G., and Wolff, G. J. (1993). Optimal brain surgeon and general network pruning. In *Proc. IEEE International Conference on Neural Networks*, volume 1, pages 293–299.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning.* Springer.

[Haykin, 1998] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation.* Prentice Hal, 2nd edition.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis for speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752.

[Hermansky and Morgan, 1994] Hermansky, H. . and Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589.

[Hermansky et al., 2000] Hermansky, H., Ellis, D., and Sharma, S. (2000). Tandem connectionist feature stream extraction for conventional HMM systems. In *Proc. IEEE ICASSP*, pages 1635–1638, Istanbul, Turkey.

[Hermansky and Sharma, 1998] Hermansky, H. and Sharma, S. (1998). TRAPs - classifiers of temporal patterns. In *Proc. ICSLP.*

[Hifny, 2002] Hifny, Y. (2002). Generalized improved iterative scaling for random field parameters estimation. http://www.dcs.shef.ac.uk/ yhifny/publications/giis.pdf.

[Hifny and Renals, 2005] Hifny, Y. and Renals, S. (2005). Acoustic modelling based on conditional random field. In *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, UK.

[Hifny et al., 2005] Hifny, Y., Renals, S., and Lawrence, N. (2005). A hybrid Max-Ent/HMM based ASR system. In *Proc. INTERSPEECH*, pages 3017–3020, Lisbon, Portugal.

[Hifny et al., 2004] Hifny, Y., Renals, S., and Lawrence, N. D. (2004). Acoustic space dimensionality selection and combination using the maximum entropy principle. In *Proc. IEEE ICASSP*, volume 5, pages 637–640, Montreal, Canada.

[Ho and Pepyne, 2002] Ho, Y. and Pepyne, D. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570.

[Hochreiter and Schmidhuber, 1999] Hochreiter, S. and Schmidhuber, J. (1999). Feature extraction through LOCOCODE. *Neural Computation*, 11(3):679–714.

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

[Hong-Kwang and Gao, 2003] Hong-Kwang, J. K. and Gao, Y. (2003). Maximum entropy direct models for speech recognition. In *Proc IEEE ASRU Workshop*, pages 1– 6, St. Thomas, U.S. Virgin Islands.

[Huang et al., 2001] Huang, X., Acero, A., and Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall.

[Hunt et al., 1991] Hunt, M., Richardson, S., Bateman, D., and Piau, A. (1991). An investigation of PLP and IMELDA acoustic representations and of their potential for combination. In *Proc. IEEE ICASSP*, volume 2, pages 881–884, Toronto, Canada.

[Hwang and Huang, 1991] Hwang, M. and Huang, X. (1991). Acoustic classification of phonetic hidden Markov models. In *Proc. EUROSPEECH*, Genova, Italy.

[Jaakkola and Haussler, 1998] Jaakkola, T. S. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *Proc. NIPS*, volume 11.

[Jacobs, 1988] Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307.

[Jaynes, 1957] Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4):620–630.

[Jaynes, 1982] Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proc. of IEEE*, 70(9):939–952.

[Jaynes, 2003] Jaynes, E. T. (2003). *Probability Theory: The Logic of Science.* Cambridge University Press.

[Jebara, 2002] Jebara, T. (2002). *Discriminative, Generative, and Imitative Learning.* PhD thesis, Massachusetts Institute of Technology.

[Jelinek, 1997] Jelinek, F. (1997). *Statistical Methods for Speech Recognition.* MIT Press.

[Joachims, 1998] Joachims, T. (1998). Making large-scale support vector machine learning practical. In Scholkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT Press.

[Jordan and Sejnowski, 2001] Jordan, M. and Sejnowski, T. J., editors (2001). *Graphical Models: Foundations of Neural Computation.* MIT Press.

[Juang and Katagiri, 1992] Juang, B. and Katagiri, S. (1992). Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12):3043– 3054.

[Kaiser et al., 2002] Kaiser, J., Horvat, B., and Ka, Z. (2002). Overall risk criterion estimation of hidden Markov model parameters. *Speech Communication*, 38(3-4):383–398.

[Kanevsky, 2004] Kanevsky, D. (2004). Extended Baum transformations for general functions. In *Proc. IEEE ICASSP*, volume 5.

[Kapadia, 1998] Kapadia, S. (1998). *Discriminative Training of Hidden Markov Models.* PhD thesis, University of Cambridge.

[Kapadia et al., 1993] Kapadia, S., Valtchev, V., and Young, S. (1993). MMI training for continuous phoneme recognition on the TIMIT database. In *Proc. IEEE ICASSP*, volume 2, pages 491–494, Minneapolis, MN, USA.

[Kapur and Kesavan, 1992] Kapur, J. and Kesavan, H. (1992). *Entropy Optimization Principles with Applications.* Academic Press.

[Katagiri et al., 1998] Katagiri, S., Juang, B.-H., and Lee, C.-H. (1998). Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. *Proc. IEEE*, 11:2345–2373.

[Kershaw et al., 1996] Kershaw, D., Robinson, T., and Renals, S. (1996). The 1995 Abbot hybrid connectionist-HMM large vocabulary recognition system. In *Proc. ARPA Spoken Language Technology Conference*, pages 93–99.

[Kinderman and Snell, 1980] Kinderman, R. and Snell, J. L. (1980). *Markov Random Fields and their Applications.* American Mathematical Society, Providence, RI.

[Kingsbury et al., 1998] Kingsbury, B., Morgan, N., and S.Greenberg (1998). Robust speech recognition using the modulation spectrogram. *Speech Communication*, 25(1-3):117–132.

[Kingsbury and Morgan, 1997] Kingsbury, B. E. and Morgan, N. (1997). Recognizing reverberant speech with RASTA-PLP. In *Proc. IEEE ICASSP*, volume 2, pages 1259–1262, Munich, Germany.

[Kirchhoff, 1998] Kirchhoff, K. (1998). Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments. In *Proc. ICSLP*, pages 891–894, Sydney, Australia.

[Kreel, 1998] Kreel, U. H.-G. (1998). Pairwise classification and support vector machines. In Scholkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods: Support Vector Learning*, pages 255–268. MIT Press.

[Krogh and Riis, 1999] Krogh, A. and Riis, S. K. (1999). Hidden neural networks. *Neural Computation*, 11(2):541–563.

[Kumar and Andreou, 1998] Kumar, N. and Andreou, A. G. (1998). Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communcation*, 26(4):283–297.

[Lafferty, 2002] Lafferty, J. (2002). Personal communication.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289.

[Lafferty et al., 2004] Lafferty, J., Zhu, X., and Liu, Y. (2004). Kernel conditional random fields: representation and clique selection. In *Proc. ICML*, Banff, Alberta, Canada.

[Lauritzen, 1996] Lauritzen, S. (1996). *Graphical Models*. Oxford University Press.

[Layton, 2004] Layton, M. (2004). Maximum margin training of generative kernels. Technical Report CUED/F-INFENG/TR.484, Cambridge University.

[LeCun et al., 1998a] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998a). Gradient-based learning applied to document recognition. *Proc. of IEEE*, 86(11):2278–2324.

[LeCun et al., 1998b] LeCun, Y., Bottou, L., Orr, G. B., and Mueller, K.-R. (1998b). Efficient backprop. In *Neural Networks—Tricks of the Trade, Lecture Notes in Computer Sciences 1524*, pages 5–50. Springer.

[LeCun et al., 1990] LeCun, Y., Denker, J., and Solla, S. (1990). Optimal brain damage. In *Proc. NIPS*, volume 2, pages 598–605.

[Lee, 1988] Lee, K.-F. (1988). *Large Vocabulary Speaker-independent Continuous Speech Recognition: The SPHINX System*. PhD thesis, Carnegie Mellon University.

[Lee and Hon, 1989] Lee, K.-F. and Hon, H.-W. (1989). Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 37(11):1641–1648.

[Lee and Rose, 1996] Lee, L. and Rose, C. (1996). Speaker normalisation using efficient frequency warping procedures. In *Proc. IEEE ICASSP*, volume 1, pages 353–356, Atlanta, GA, USA.

[Levenshtein, 1965] Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848. (Russian). English translation in Soviet Physics Doklady, 10(8):707-710, 1966.

[Levinson, 1986] Levinson, S. E. (1986). Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):29–45.

[Li et al., 2006] Li, J., Yuan, M., and Lee, C.-H. (2006). Soft margin estimation of hidden Markov model parameters. In *Proc. INTERSPEECH*, page 2422 2425, Pennsylvania, USA.

[Li et al., 2005] Li, X., Jiang, H., and Liu, C. (2005). Large margin HMMs for speech recognition. In *Proc. IEEE ICASSP*, volume 5, pages 513– 516, Philadelphia.

[Likhododev and Gao, 2002] Likhododev, A. and Gao, Y. (2002). Direct models for phoneme recognition. In *Proc. IEEE ICASSP*, volume 1, pages 89–92, Orlando, FL, USA.

[Lippmann, 1997] Lippmann, R. (1997). Speech recognition by machines and humans. *Speech Communcation*, 22(1):1–16.

[Liu et al., 2005] Liu, C., Jiang, H., and Li, X. (2005). Discriminative training of CDHMMs for maximum relative separation margin. In *Proc. IEEE ICASSP*, volume 1, pages 101– 104, Philadelphia.

[Macherey et al., 2005] Macherey, W., Haferkamp, L., Schlöuter, R., and Ney, H. (2005). Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In *Proc. INTERSPEECH*, pages 2133–2136, Lisbon, Portugal.

[Macherey and Ney, 2003] Macherey, W. and Ney, H. (2003). A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition. In *Proc. EUROSPEECH*, pages 493–496, Geneva, Switzerland.

[MacKay, 2003] MacKay, D. (2003). *Information Theory, Pattern Recognition, and Neural Networks.* Cambridge University Press.

[Mahajan et al., 2006] Mahajan, M., Gunawardana, A., and Acero, A. (2006). Training algorithms for hidden conditional random fields. In *Proc. IEEE ICASSP*, volume 1, pages 273–276, Toulouse, France.

[Malouf, 2002] Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proc. CoNLL*, pages 49–55.

[Mangu et al., 1999] Mangu, L., Brill, E., and Stolcke, A. (1999). Finding consensus among words: Lattice-based word error minimization. In *Proc. EUROSPEECH*, pages 495–498, Budapest, Hungary.

[Mangu et al., 2000] Mangu, L., Brill, E., and Stolcke, A. (2000). Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400.

[Moore, 1991] Moore, A. (1991). A tutorial on kd-trees. Extract from PhD thesis. Available from http://www.autonlab.org/autonweb/14665.html.

[Morgan and Bourlard, 1995] Morgan, N. and Bourlard, H. (1995). Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach. *IEEE Signal Processing Magazine*, 12(3):25–42.

[Morgan et al., 2005] Morgan, N., Qifeng, Z., Stolcke, A., Sonmez, K., Sivadas, S., Shinozaki, T., Ostendorf, M., Jain, P., Hermansky, H., Ellis, D., Doddington, G., Chen, B., Cretin, O., Bourlard, H., and Athineos, M. (2005). Pushing the envelope - aside. *IEEE Signal Processing Magazine*, 22(5):81– 88.

[Murata et al., 1997] Murata, N., Müller, K.-R., Ziehe, A., and ichi Amari, S. (1997). Adaptive on-line learning in changing environments. In *Proc. NIPS*, volume 9, page 599.

[Na et al., 1995] Na, K., Jeon, B., Chang, D., Chae, S., and Ann, S. (1995). Discriminative training of hidden Markov models using overall risk criterion and reduced gradient method. In *Proc. EUROSPEECH*, pages 97–100, Madrid, Spain.

[Nadas, 1983] Nadas, 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(4):814–817.

[Nocedal and Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization.* Springer.

[Normandin, 1991] Normandin, Y. (1991). *Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem.* PhD thesis, McGill University.

[Normandin et al., 1994] Normandin, Y., Lacouture, R., and Cardin, R. (1994). MMIE training for large vocabulary continuous speech recognition. In *Proc. ICSLP*, pages 1367–1370, Yokohama, Japan.

[Okawa et al., 1998] Okawa, S., Bocchieri, E., and Potamianos, A. (1998). Multiband speech recognition in noisy environments. In *Proc. IEEE ICASSP*, volume 2, pages 641–644, Seattle, Washington, USA.

[Okawa et al., 1999] Okawa, S., Nakajima, T., and Shirai, K. (1999). A recombination strategy for multi-band speech recognition based on mutual information criterion. In *Proc. EUROSPEECH*, pages 603–606, Budapest, Hungary.

[Omar and Hasegawa-Johnson, 2003] Omar, M. and Hasegawa-Johnson, M. (2003). Approximately independent factors of speech using nonlinear symplectic transformation. *IEEE Transactions on Speech and Audio Processing*, 11(6):660–671.

[Omohundro, 1987] Omohundro, S. M. (1987). Efficient algorithms with neural network behaviour. *Journal of Complex Systems*, 1(2):273–347.

[Ostendorf et al., 1996] Ostendorf, M., Digalakis, V., and Kimball, O. (1996). From HMMs to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 4(5):360–378.

[Osuna et al., 1997] Osuna, E., Freund, R., and Girosi, F. (1997). Improved training algorithm for support vector machines. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 276–285.

[Paliwal et al., 1995] Paliwal, K., Bacchiani, M., and Sagisaka, Y. (1995). Minimum classification error training algorithm for feature extractor and pattern classifier in speech recognition. In *Proc. EUROSPEECH*, volume 1, pages 541–544, Madrid, Spain.

[Parzen, 1961] Parzen, E. (1961). An approach to time series analysis. *Annals of Mathematical Statistics*, 32:951–989.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc.

[Perkins et al., 2003] Perkins, S., Lacker, K., and Theiler, J. (2003). Grafting: fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356.

[Platt, 2000] Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Smola, A., Bartlett, P., Schoelkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classiers*, pages 61–74.

[Platt et al., 2000] Platt, J., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin DAGS for multiclass classification. In *Proc. NIPS*, volume 12, pages 547–553.

[Povey, 2004] Povey, D. (2004). *Discriminative Training for Large Vocabulary Speech Recognition.* PhD thesis, University of Cambridge.

[Povey, 2005] Povey, D. (2005). Improvements to fMPE for discriminative training of features. In *Proc. INTERSPEECH*, pages 2977–2980, Lisbon, Portugal.

[Povey et al., 2005] Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., and Zweig, G. (2005). fMPE: Discriminatively trained features for speech recognition. In *Proc. IEEE ICASSP*, volume 1, pages 961– 964, Philadelphia.

[Povey and Woodland, 2002] Povey, D. and Woodland, P. (2002). Minimum phone error and I-smoothing for improved discriminative training. In *Proc. IEEE ICASSP*, volume 1, pages 105–108, Orlando, FL.

[Powell, 1987] Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. In Mason and Cox, M., editors, *Algorithms for approximation*, pages 143–167. Oxford University Press.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286.

[Rabiner and Juang, 1993] Rabiner, L. R. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition.* Prentice Hall.

[Rahim and Lee, 1996] Rahim, M. and Lee, C. H. (1996). Simultaneous ANN feature and HMM recognizer design using string-based Minimum Classification Error (MCE) training. In *Proc. ICSLP*, Philadelphia, PA.

[Ratnaparkhi, 1996] Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proc. EMNLP*, pages 133–142.

[Ratnaparkhi, 1997] Ratnaparkhi, A. (1997). A simple introduction to maximum entropy models for natural language processing. Technical Report IRCS-08-97, Institute for Research in Cognitive Science, University of Pennsylvania.

[Reichl and Ruske, 1995] Reichl, W. and Ruske, G. (1995). Discriminative training for continuous speech recognition. In *Proc. EUROSPEECH*, volume 1, pages 537–540, Madrid, Spain.

[Richards and Bridle, 1999] Richards, H. and Bridle, J. (1999). The HDM: A segmental hidden dynamic model of coarticulation. In *Proc. IEEE ICASSP*, pages 357–360, Phoenix, Arizona.

[Riedmiller and Braun, 1993] Riedmiller, M. and Braun, H. (1993). A direct method for faster backpropagation learning: The RPROP algorithm. In *Proc. IEEE International Conference on Neural Networks*, pages 586–591.

[Rissanen, 2005] Rissanen, J. (2005). An introduction to the MDL principle. *http://www.mdl-research.org/jorma.rissanen/pub/Intro.pdf.*

[Robinson, 1994] Robinson, A. (1994). An application of recurrent neural nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305.

[Robinson, 1989] Robinson, A. J. (1989). *Dynamic Error Propagation Networks.* PhD thesis, University of Cambridge.

[Rosenfeld, 1994] Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* PhD thesis, Carnegie Mellon University.

[Roweis and Ghahramani, 1999] Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345.

[Rumelhart et al., 1986] Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In Rumelhart, D. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press.

[Saon and Padmanabhan, 2000] Saon, G. and Padmanabhan, M. (2000). Minimum Bayes error feature selection for continuous speech recognition. In *Proc. NIPS*, volume 13, pages 800–806.

[Saon et al., 2000] Saon, G., Padmanabhan, M., Gopinath, R., and Chen, S. (2000). Maximum likelihood discriminant feature spaces. In *Proc. IEEE ICASSP*, volume 2, pages 1129–1132, Istanbul, Turkey.

[Schlkopf et al., 1998] Schlkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319.

[Schlüter et al., 1997] Schlüter, R., Macherey, W., Kanthak, S., Ney, H., and Welling, L. (1997). Comparison of optimization methods for discriminative training criteria. In *Proc. EUROSPEECH*, pages 15–18, Rhodes, Greece.

[Schlüter et al., 2001] Schlüter, R., Macherey, W., Müller, B., and Ney, H. (2001). Comparison of discriminative training criteria and optimization methods for speech recognition. *Speech Communication*, 34(3):287–310.

[Scholkopf and Smola, 2002] Scholkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press.

[Schraudolph, 1999] Schraudolph, N. N. (1999). Local gain adaptation in stochastic gradient descent. Technical Report IDSIA-09-99, IDSIA.

[Schraudolph, 2002] Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738.

[Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

[Schwarz et al., 2004] Schwarz, P., Matejka, P., and Cernocky, J. (2004). Towards lower error rates in phoneme recognition. In *Proc. TSD2004*, pages 465–472.

[Sha and Pereira, 2003] Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. Technical Report CIS TR MS-CIS-02-35, University of Pennsylvania.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

[Shewchuk, 1994] Shewchuk, J. (1994). An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University.

[Smith and Gales, 2002] Smith, N. and Gales, M. (2002). Speech recognition using SVMs. In *Proc. NIPS*, volume 14.

[Smith et al., 2001] Smith, N., Gales, M., and Niranjan, M. (2001). Data dependent kernels in SVM classification of speech patterns. Technical Report CUED/F-INFENG/TR.387, University of Cambridge.

[Smyth et al., 1997] Smyth, P., Heckerman, D., and Jordan, M. I. (1997). Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269.

[Sutton, 1992] Sutton, R. S. (1992). Adapting bias by gradient descent: An incremental version of delta-bar-delta. In *Proc. AAAI*, pages 171–176.

[Tokuda et al., 2004] Tokuda, K., Zen, H., and Kitamura, T. (2004). Reformulating the HMM as a trajectory model. In *Proc. of Beyond HMM – Workshop on statistical modeling approach for speech recognition.*

[Tóth and Kocsor, 2005] Tóth, L. and Kocsor, A. (2005). On naive Bayes in speech recognition. *Int. Journal of Applied Mathematics and Computer Science*, 15(2):287–294.

[Trentin and Gori, 2001] Trentin, E. and Gori, M. (2001). A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, 37(1-4):91–126.

[Tritschler and Gopinath, 1999] Tritschler, A. and Gopinath, R. (1999). Improved speaker segmentation and segments clustering using the Bayesian information criterion. In *Proc. EUROSPEECH*, pages 679–682.

[Valtchev et al., 1997] Valtchev, V., Odell, J. J., Woodland, P. C., and Young, S. J. (1997). MMIE training of large vocabulary speech recognition systems. *Speech Communication*, 22(4):303–314.

[Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley-Interscience.

[Vishwanathan et al., 2006] Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M. W., and Murphy, K. P. (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *Proc. ICML*, pages 969–976.

[Wahba, 1990] Wahba, G. (1990). *Spline Models for Observational Data.* SIAM: Society for Industrial and Applied Mathematics.

[Wahba, 1999] Wahba, G. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. In Scholkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods: Support Vector Learning*, pages 69–88. MIT Press.

[Wang et al., 2002] Wang, S., Schuurmans, D., and Zhao, Y. (2002). The latent maximum entropy principle. *submitted to IEEE Transactions on Information Theory.*

[Welling et al., 1999] Welling, L., Kanthak, S., and Ney, H. (1999). Improved methods for vocal tract normalization. In *Proc. IEEE ICASSP*, pages 761–764, Phoenix, Arizona.

[Weston and Watkins, 1998] Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London.

[Woodland and Povey, 2000] Woodland, P. and Povey, D. (2000). Large scale discriminative training for speech recognition. In *ISCA ITRW Automatic Speech Recognition: Challenges for the Millenium*, pages 7–16.

[Young, 1996] Young, S. (1996). A review of large-vocabulary continuous-speech. *IEEE Signal Processing Magazine*, 13(5):45–57.

[Young et al., 2001] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2001). *The HTK Book, Version 3.1.*

[Young and Woodland, 1994] Young, S. and Woodland, P. (1994). State clustering in HMM-based continuous speech recognition. *Computer Speech and Language*, 8(4):369–384.

[Zhu and Hastie, 2001] Zhu, J. and Hastie, T. (2001). Kernel logistic regression and the import vector machine. In *Proc. NIPS*, volume 13.

[Zweig and Russell, 1998] Zweig, G. and Russell, S. J. (1998). Speech recognition with dynamic Bayesian networks. In *Proc. AAAI*, pages 173–180.