Discriminative models for speech recognition

Anton Ragni

Peterhouse

University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

2013

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. It has not been submitted in whole or in part for a degree at any other university. Some of the work has been published previously in the form of conference proceedings [191, 192, 242], journal articles [278], workshop papers [72, 190] and technical reports [241]. The length of this thesis including appendices, bibliography, footnotes, tables and equations is approximately 43,000 words. This thesis contains 27 figures and 22 tables.

Abstract

The discriminative approach to speech recognition offers several advantages over the generative, such as a simple introduction of additional dependencies and direct modelling of sentence posterior probabilities/decision boundaries. However, the number of sentences that can possibly be encoded into an observation sequence can be vast, which makes the application of models, such as support vector machines, difficult in speech recognition tasks. To overcome this issue, it is possible to apply acoustic code-breaking in order to decompose the whole-sentence recognition problem into a sequence of independent word recognition problems. However, the amount of training data that is usually available provides sufficient coverage for only a small number of most frequent words. Alternatively, a related solution from the generative approach can be adopted, where decomposition into sub-sentence units is introduced directly into the model. There have been previously proposed decompositions into words, phones and states. Among those approaches, the decomposition into phones retains sufficiently long-span dependencies and good coverage in the training data. However, in order to make it more generally applicable, the word- and phone-level modelling need to be combined. In addition, the use of context-dependent phones useful for large vocabulary tasks need to be investigated.

The first contribution of this thesis is extended acoustic code-breaking, where the training data insufficiency problem is addressed by synthesising examples for under-represented words.

The second contribution of this thesis is a discriminative model that combines context-dependent phone-level acoustic modelling with word-level language and pronunciation modelling.

Acknowledgements

The author would like to express his deep gratitude to Toshiba Research Europe Limited, Engineering and Physical Sciences Research Council, Hidden Markov Model Toolkit Project for providing funding and Mark Gales for supervising. The author would also like to express his sincere appreciation to past, current and future colleagues at University of Tartu, Nordic Graduate School of Language Technologies, Toshiba Research Europe Limited, University of Cambridge. Finally, the author would like to thank his family and friends for their support.

Contents

List of Figures

List of Tables

Nomenclature

Chapter 1

$\mathbf{o}$	observation vector
$\mathbf{O}_{1:T}$	observation sequence
ω	sentence
T	number of observations
t	time/observation index

Chapter 2

a	arc
$a_{i,j}$	probability of making transition from state S_i to state S_j
α_a	arc forward probability
$\alpha_j(t)$	forward probability
α'_a	arc forward correctness
$a_{\mathcal{F}}$	sentence end arc
$a_{\mathcal{I}}$	sentence beginning arc
$a^{\mathtt{i}}$	arc identity
$b_j(\cdot)$	state output distribution

$\mathbb{L}$	lattice
$\mathbb{L}_{\texttt{den}}$	denominator lattice
$\mathbb{L}_{\texttt{num}}$	numerator lattice
$[[\mathbb{L}]]$	lattice weight/cost
β_a	arc backward probability
$\beta_j(t)$	backward probability
β'_a	arc backward correctness
$\mathbf{a}$	arc sequence
$\boldsymbol{\lambda}$	HMM parameters
$\boldsymbol{\mu}_{j,m}$	mean vector
$\boldsymbol{\mu}^{\texttt{p}}_{j,m}$	prior mean
$\boldsymbol{\Sigma}^{\texttt{p}}_{j,m}$	prior covariance
$\mathbf{O}^{\texttt{s}}_{1:T}$	static observation sequence
$\mathbf{o}^{\texttt{s}}_{t}$	static observation vector at time t
$\mathbf{q}_{1:T}$	state sequence
$\boldsymbol{\Sigma}_{j,m}$	covariance matrix
$\boldsymbol{\Theta}_{j,m}$	covariance statistics
$\boldsymbol{\theta}_{j,m}$	mean statistics
$\boldsymbol{\Theta}^{\texttt{den}}_{j,m}$	denominator term covariance statistics
$\boldsymbol{\theta}^{\texttt{den}}_{j,m}$	denominator term mean statistics
$\boldsymbol{\Theta}^{\texttt{num}}_{j,m}$	numerator term covariance statistics
$\boldsymbol{\theta}^{\texttt{num}}_{j,m}$	numerator term mean statistics

NOMENCLATURE

$\mathbf{w}_{1:L}$	word sequence
c_a	average accuracy of arc sequences passing arc a
$c_{j,m}$	component weight
$c^{(r)}$	average accuracy of all arc sequences in $\mathbb{L}_{\mathtt{den}}^{(r)}$
$\Delta^{(1)}\mathbf{o}_t^{\mathtt{s}}$	delta coefficients
$\Delta^{(2)}\mathbf{o}_t^{\mathtt{s}}$	acceleration coefficients
γ_a	arc occupancy
$\gamma_{a,j,m}(t)$	arc state-component occupancy
$\gamma_{j,m}$	occupancy statistics
$\gamma_{j,m}^{\mathtt{den}}$	denominator term occupancy
$\gamma_{j,m}^{\mathtt{num}}$	numerator term occupancy
$\gamma_{j,m}(t)$	state-component occupancy
$\gamma_j(t)$	state occupancy
$\gamma_a^{\mathtt{mpe}}$	MPE differential
κ	acoustic de-weighting constant
$\tau^{\mathtt{I}}$	I-smoothing constant
L	number of words in a sequence
M	number of Gaussian components
$\mathcal{A}(\cdot)$	accuracy function
$\mathcal{D}$	(supervised) training data
$\mathcal{F}_{\mathtt{ml}}(\cdot)$	maximum likelihood objective function
$\mathcal{F}_{\mathtt{mmi}}(\cdot)$	MMI objective function

$\mathcal{F}_{\texttt{mpe}}(\cdot)$	MPE objective function
$\mathcal{G}_{\texttt{num}}(\cdot)$	MMI weak-sense auxiliary function
$\mathcal{G}_{\texttt{mpe}}(\cdot)$	final MPE weak-sense auxiliary function
$\mathcal{L}(\cdot)$	loss function
$\mathcal{N}(\cdot)$	multi-variate Gaussian
$\mathcal{Q}(\cdot)$	ML auxiliary function
$\mathcal{Q}_{\texttt{den}}(\cdot)$	denominator term weak-sense auxiliary function
$\mathcal{Q}_{\texttt{mpe}}(\cdot)$	MPE weak-sense auxiliary function
$\mathcal{Q}_{\texttt{num}}(\cdot)$	numerator term weak-sense auxiliary function
$\mathcal{Q}_{\texttt{sm}}(\cdot)$	smoothing function
N	number of HMM states
n	order
$\overline{\boldsymbol{\mu}}_{j,m}$	canonical/unadapted/clean mean
$\overline{\boldsymbol{\Sigma}}_{j,m}$	canonical/unadapted/clean covariance
q_t^j	event of being in state S_j at time t
$q_t^{j,m}$	event of being in state S_j and occupying component m at time t
R	number of training sequences
S_j	HMM state
w	word
$\widehat{\boldsymbol{\lambda}}$	new HMM parameters
$\widehat{\mathbf{q}}_{1:T}$	optimal state sequence
$\widetilde{\mathcal{A}}(\cdot)$	approximate phone arc accuracy

$\zeta_{i,j}(t)$	state-state occupancy

Chapter 3

$\boldsymbol{\alpha}$	discriminative model parameters
$\boldsymbol{\alpha}^{\texttt{p}}$	discriminative model prior parameters
$\boldsymbol{\alpha}^{\texttt{svm}}$	SVM parameters
$\boldsymbol{\phi}(\cdot)$	feature-function
$\delta(\cdot)$	delta function
$k(\cdot)$	kernel
$\mathcal{F}(\cdot)$	final objective function
$\mathcal{F}_{\texttt{cml}}(\cdot)$	CML objective function
$\mathcal{F}_{\texttt{lm}}(\cdot)$	large margin objective function
$\mathcal{F}_{\texttt{mbr}}(\cdot)$	MBR objective function
∇	gradient
ξ_r	slack variable
$Z(\cdot)$	normalisation term

Chapter 5

a	segment/arc
$a^{\texttt{i}}$	segment/arc identity
$\mathbf{a}$	segmentation/arc sequence
$\boldsymbol{\alpha}_{\texttt{am}}$	discriminative acoustic model parameters
$\boldsymbol{\alpha}_{\texttt{lm}}$	discriminative language model parameters
$\boldsymbol{\alpha}_{\texttt{pm}}$	discriminative pronunciation model parameters

$\{a\}$	index of observations for segment/arc
$\widehat{\mathbf{a}}$	optimal segmentation
$\widetilde{P}(\cdot)$	segment/arc transition score
$\boldsymbol{\phi}_{\mathtt{a}}$	appended likelihood score-space
$\boldsymbol{\phi}_{\mathtt{f}}$	Fisher score-space
$\boldsymbol{\phi}_{\mathtt{l}}$	likelihood score-space
$\boldsymbol{\phi}_{\mathtt{l}}^{(1)}$	first-order likelihood score-space
$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$	HMM mean derivative score-space
$\boldsymbol{\phi}_{\mathtt{r}}$	likelihood ratio score-space
$\mathcal{V}$	vocabulary of segment/arc identities
u	linguistic unit
v	vocabulary element

Acronyms

CAug	Conditional Augmented (Model)
CER	Character Error Rate
CML	Conditional Maximum Likelihood
CMLLR	Constrained Maximum Likelihood Linear Regression
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transformation
DFT	Discrete Fourier Transform
DSAT	Discriminative Speaker Adaptive Training
DVAT	Discriminative VTS Adaptive Training

NOMENCLATURE

EBW	Extended Baum-Welch
EM	Expectation Maximisation
FA	Factor Analysis
GMM	Gaussian Mixture Model
HCRF	Hidden Conditional Random Fields
HMM	Hidden Markov Model
LPC	Linear Prediction Coding Coefficients
MAP	Maximum-a-Posteriori
MaxEnt	Maximum Entropy (Model)
MBR	Minimum Bayes' risk
MEMM	Maximum Entropy Markov Model
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
ML	Maximum Likelihood
MPE	Minimum Phone Error
MWE	Minimum Word Error
PLP	Perceptual Linear Prediction Coefficients
SCRF	Segmental Conditional Random Fields
SER	Sentence Error Rate
SVM	Support Vector Machines
VAT	VTS Adaptive Training
VTS	Vector Taylor Series

WER Word Error Rate

WFST Weighted Finite-State Transducer

Chapter 1

Introduction

There are many applications involving recognition of patterns, such as speech waveforms, images and protein sequences. Examples include speech [106, 250] and speaker [58] recognition, document image analysis [48] and remote sensing [226], bioinformatics [193]. Among the various frameworks in which pattern recognition has been formulated, the statistical approach has been most intensively studied and used in practice [102]. The statistical framework has been also widely adopted for speech recognition [60, 106, 188, 263] - the problem domain of this thesis.

The statistical framework assumes that speech waveform can be represented by a sequence of observation vectors $\mathbf{O}_{1:T} = \mathbf{o}_1, \ldots, \mathbf{o}_T$ and that this sequence encodes sentence ω [263]. In order to decode the sentence, a mapping, optimal in some meaningful sense, from observation sequences to sentence identities is learnt [106, 263]. One approach is to learn an indirect mapping from the audio to the text by combining acoustic, $p(\mathbf{O}_{1:T}|\omega)$, and language, $P(\omega)$, models using Bayes' rule[1] to yield sentence $\widehat{\omega}$ maximising a-posteriori probability [263]

$$\widehat{\omega} = \arg\max_{\omega}\{p(\mathbf{O}_{1:T}|\omega)P(\omega)\} \tag{1.1}$$

These models are usually called generative models since by sampling from the acoustic and language model it is possible to generate synthetic examples of the observation sequences and sentences [17].[2] This is the basis of a *generative ap-*

[1] According to Bayes' rule, $P(\omega|\mathbf{O}_{1:T}) = \frac{p(\mathbf{O}_{1:T}|\omega)P(\omega)}{p(\mathbf{O}_{1:T})}$, where $p(\mathbf{O}_{1:T})$ is constant for all ω.

[2] Chapter 4 discusses several approaches how the observation sequences can be sampled.

proach to speech recognition [89]. Another approach is to learn a direct map from the audio to the text, for instance, using a direct model of posterior probability, $P(\omega|\mathbf{O}_{1:T})$, which yields sentence $\widehat{\omega}$ maximising the a-posteriori probability [127]

$$\widehat{\omega} = \arg\max_{\omega}\{P(\omega|\mathbf{O}_{1:T})\} \tag{1.2}$$

Direct models are usually called discriminative models since they directly discriminate between the different values of ω [17]. This is the basis of a *discriminative approach* to speech recognition [89]. For speech recognition applications, both approaches must be able to handle the variable-length nature of observation sequences and vast number of possible sentences, model dependencies in the observation sequence and stay robust to speaker and noise conditions [70, 73, 75].

The generative approach to speech recognition discussed in Chapter 2 is based on a beads-on-a-string model, so called because all sentences are represented by concatenating a sequence of precomputed sub-sentence, such as word or phone, models together [263]. These sub-sentence models are hidden Markov models (HMM) [188], an example of sequence models [73, 224], capable of handling variable-length observation sequence. This is the basis of HMM-based approach to speech recognition. The major issue that is often cited with the HMM-based approach are conditional independence assumptions underlying the HMM [75, 233, 234, 270]. In particular, the individual observations are assumed to be independent given the hidden states that generated them [75]. Although these assumptions are the key to efficient parameter estimation from large quantities of training data and decoding with these models [188], they severely limit the range of possible dependencies that can be modelled and are believed to be partly responsible for unsatisfactory performance in many situations [72].

The HMM-based approaches have significantly evolved over the years [75, 188]. In particular, modern HMM-based speech recognition systems usually adopt a multi-pass architecture, where the sentence $\widehat{\omega}$ in equation (1.1) is determined in stages [75]. The first stage in the multi-pass architecture usually consists of producing a large number of hypothesised sentence identities in a compact lattice format. The lattice, compared to solving the problem in equation (1.1) in a single pass, can be efficiently re-scored using more advanced acoustic and language models [265] or converted into a *confusion network* that decomposes encoded hypotheses into a sequence of independent binary or multi word confusions [52, 150] for a more elaborate decoding [26, 54, 218] or to provide confidence indication about reliability of the hypothesised sentence [88].

The discriminative approach to speech recognition discussed in Chapter 3, 4, 5 and 6 have not been previously investigated for tasks other than smaller vocabulary systems because of the complexity associated with learning direct maps from observation sequences to sentences [75]. In particular, handling variable-length observation sequences and vast number of possible sentences with the discriminative models, such as maximum entropy model (MaxEnt) and support vector machines (SVM), is complicated. A number of approaches in the form of *feature-functions* have been proposed to map variable-length observation sequences into fixed-dimensional feature space. In order to handle the vast number of possible sentences, two major solutions have been proposed. The first solution, *acoustic code-breaking* [68, 128, 245], consists of decomposing the whole-sentence recognition problem into a sequence of *independent* word recognition problems, for instance, using confusion networks. The word recognition problems then can be independently addressed by using MaxEnt [278] and SVM [247] classifiers. Compared to the HMM, this solution reduces the range of dependencies possible to model from the sentence to word level. The second solution, *structured discriminative models* [85, 127, 128, 281], adopts a similar to the beads-on-a-string representation, where all sentences are represented by concatenating a sequence of precomputed sub-sentence models together [73]. Compared to acoustic code-breaking, this solution provides a model of the entire sentences. A range of structured discriminative models have been proposed. Some of them, such as maximum entropy Markov models (MEMM), have structures similar to the HMM [127]. Others, such as segmental conditional random fields (SCRF), apply conditional independence assumptions at the word segment level [281]. Compared to the HMM and MEMM, these models reduce the range of dependences possible to model from the sentence to word segment level, similar to the acoustic code-breaking.

One limitation of acoustic code-breaking is that it can not be applied in situations where limited or no examples of the words exists in the training data. This has limited previous applications to re-scoring only a small number of the most frequently occurring word-pair confusions. The first contribution of this thesis is *extended acoustic code-breaking*, where the training data for under represented words is artificially generated. Here, a simplified form of speech synthesis is sufficient, where the observation sequences rather than waveforms are required. Thus, many of the issues commonly associated in speech synthesis with waveform generation, such as excitation and prosody [166], are not relevant to this approach.

One limitation of word-level structured discriminative models is that the training data do not provide enough coverage of all words. This can make robust parameter estimation with these models complicated. One solution to this problem would be to adopt the extended acoustic code-breaking to generate data for complete sentences. However, this has not been investigated in this thesis. Another solution is to adopt a phone-level structured discriminative model [128]. Although these models reduce the range of dependencies possible to model from the word to phone level, the use of phone-level discriminative acoustic models is believed to be more appropriate for medium-to-large vocabulary speech recognition [191]. The previous work with those models have considered small vocabulary tasks based on words [278] or monophones [128]. For larger vocabulary tasks, it is common to adopt context-dependent phones to systematically address variation caused by co-articulation, stress and other factors [144]. The second contribution of this thesis are context-dependent phone structured discriminative models. In order to address large number of possible context-dependent phones and limited amounts of training data, the use of model-level phonetic decision trees clustering is proposed for tying context-dependent phone parameters.

This rest of this thesis is split into 8 chapters. A brief chapter-by-chapter breakdown is given below.

Chapter 2 provides an overview of the HMM-based approach to speech recognition. In particular it discusses the HMM and its conditional independence assumptions, beads-on-a-string modelling, approaches available for HMM parameter estimation, decoding and lattice generation, adaptation to speaker and noise conditions. In addition, several approaches to language modelling are discussed.

Chapter 3 is the first chapter discussing the discriminative approach to speech recognition. An overview of the standard, unstructured, discriminative models including the MaxEnt and SVM are given. A simple example demonstrates how variable-length observation sequences can be handled with these models.

Chapter 4 is split into two parts. The first part discusses acoustic code-breaking and details two approaches to how the whole-sentence recognition problem can be decomposed into a sequence of independent word recognition problems. The second part introduces extended acoustic code-breaking to address situations where limited or no examples of the words exist in the training data.

Chapter 5 provides an overview of structured discriminative models. In particular, it discusses the structures or beads-on-a-string representations adopted with the MEMM, CAug and SCRF models, handling of hidden variables, parameter estimation and adaptation to speaker and noise conditions.

Chapter 6 is the last chapter discussing the discriminative approach to speech recognition. An overview of feature-functions proposed for handling variable-length sequences is given. In particular, a powerful form based on generative models, such as the HMM, is discussed.

Chapter 7 introduces context-dependent phone CAug models. In particular, it discusses how context-dependent phone parameters can be tied to ensure robustness of the estimates, how feature-functions based on generative models can be applied and how the underlying generative models can be re-estimated to yield more powerful forms of CAug models.

Chapter 8 provides experimental verification to the extended acoustic code-breaking and CAug models on three speech recognition tasks where vocabulary ranges from small to medium-to-large.

Chapter 9 concludes with a summary of the thesis and suggestions for future work with the extended acoustic code-breaking and CAug models.

Chapter 2

Generative approach to speech recognition

As noted in Chapter 1, the generative approach to speech recognition is based on combining acoustic and language models to produce hypothesised sentence. This is reflected in the architecture of a typical speech recognition system in Figure 2.1 [75]. The first stage in Figure 2.1, called feature extraction, is responsible for preprocessing speech to yield an observation sequence. Given observation sequence, the second stage in Figure 2.1 employs decoder supplied with the acoustic and language model to produce the hypothesised sentence.

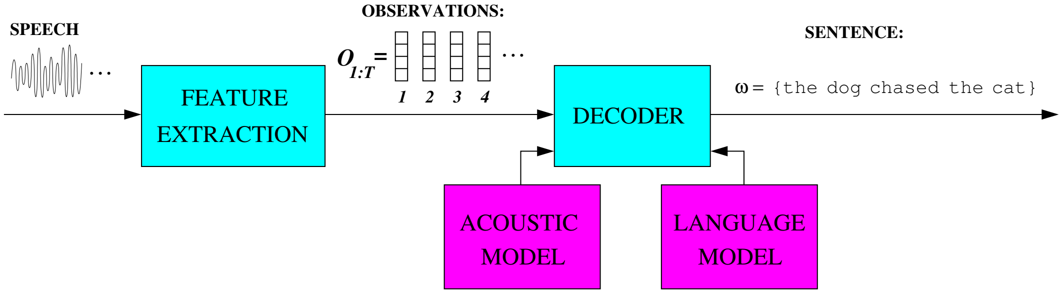


Figure 2.1: An architecture for generative approach to speech recognition

As noted in Chapter 1, different observation sequences have different lengths. In order to handle variable-length sequences, this chapter discusses sequence models [73, 224], in particular, hidden Markov models (Section 2.2). As noted in Chapter 1, the whole-sentence modelling becomes quickly impractical as the number of possible sentences increases. In order to address this issue, a structure, such as the beads-on-a-string representation [263], can be introduced into the hidden Markov model (Section 2.3). The following three sections discuss several other practical aspects including parameter tying (Section 2.4), language modelling (Section 2.5), decoding and lattice generation (Section 2.6), discriminative parameter estimation (Section 2.7) and adaptation (Section 2.8).

2.1 Observations

A range of schemes exist to extract observations from the speech signal such as linear prediction coding coefficients (LPC) [151] based on linear prediction analysis [148], Mel-frequency cepstral coefficients (MFCC) [41] based on spectral and homomorphic analyses [175, 176] and perceptual linear prediction coefficients (PLP) [92] based on linear prediction and spectral analyses. Common to all these scheme is a transformation of speech into observation sequence

$$\mathbf{O}_{1:T} = \mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T \tag{2.1}$$

where T is the number of observations, $1 : T$ is the index range $1, \ldots, T$. The feature extraction stage aims to retain in the observation sequence the salient properties of the speech whilst compressing the latter [189]. The rest of this section provides brief details on the MFCC scheme adopted in this thesis.[1]

2.1.1 Mel-frequency cepstral coefficients

The feature extraction stage based on MFCCs is illustrated by Figure 2.2. Note that prior to performing feature extraction it is common to apply a pre-emphasis (not shown in Figure 2.2) in order to spectrally flatten the speech signal [265]. The procedure can be summarised in 6 steps as follows.

(1) Group samples into blocks of 10 ms each. The length of blocks in tens of millisecond is motivated by a quasi-stationarity of vocal tract, responsible

[1] MFCCs are also utilised by a vector Taylor series noise compensation described in Section 2.8 and adopted in the experiments reported in Chapter 8.

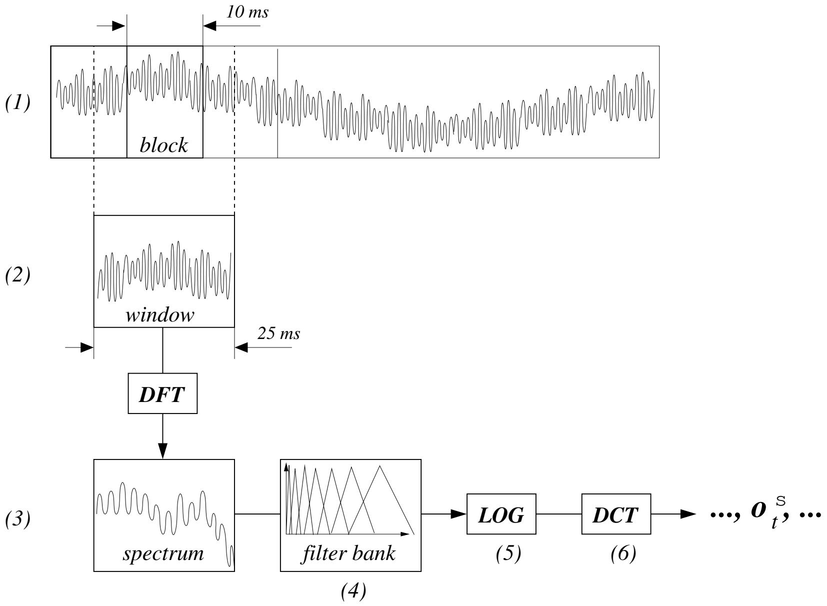


Figure 2.2: Feature extraction based on Mel-frequency cepstral coefficients

for producing speech, at these time intervals [206].

(2) Form a window of samples of 25 ms by appending the samples of adjacent blocks: $t-1$ and $t+1$. The use of overlapping windows as the units of analysis is adopted to provide smoothing between vectors of feature coefficients [189]. Furthermore, in order to minimise the adverse effect of discontinuities at the window boundaries, a smoothing window, such as Hamming window [189], is commonly applied [188].

(3) Apply a discrete Fourier transform (DFT) to compute spectrum [96].

(4) Pass the spectrum through a filter bank, where bins are spaced according to a Mel-scale [41], which approximates the frequency response of the human ear [75].

(5) Transform the output of filter bank into logarithmic domain [96].

(6) Apply a discrete cosine transformation (DCT) to yield the MFCC observation vector $\mathbf{o}_t^{\mathsf{s}}$. The goal of the DCT is to approximately de-correlate feature vector elements so that diagonal rather than full covariance matrices can be used in hidden Markov models [96, 99].

The feature extraction then proceeds to the next block, where the steps two to six are repeated. At the end of the feature extraction stage the complete MFCC observation sequence is obtained

$$\mathbf{O}^{\mathsf{s}}_{1:T} = \mathbf{o}^{\mathsf{s}}_1, \mathbf{o}^{\mathsf{s}}_2, \ldots, \mathbf{o}^{\mathsf{s}}_T \tag{2.2}$$

2.1.2 Dynamic coefficients

In sequence models such as the HMM, the use of additional, *dynamic*, information was found [59] to be advantageous to compensate for the conditional independence assumptions made by these models (Section 2.2) [75]. The dynamic information typically comes in the form of order n regression coefficients [265]

$$\begin{aligned}\Delta^{(1)}\mathbf{o}^{\mathsf{s}}_t &= \frac{1}{2\sum_{\tau=1}^{D}\tau^2}\sum_{\tau=1}^{D}\tau(\mathbf{o}^{\mathsf{s}}_{t+\tau} - \mathbf{o}^{\mathsf{s}}_{t-\tau})\\ &\vdots\\ \Delta^{(n)}\mathbf{o}^{\mathsf{s}}_t &= \frac{1}{2\sum_{\tau=1}^{D}\tau^2}\sum_{\tau=1}^{D}\tau(\Delta^{(n-1)}\mathbf{o}^{\mathsf{s}}_{t+\tau} - \Delta^{(n-1)}\mathbf{o}^{\mathsf{s}}_{t-\tau})\end{aligned} \tag{2.3}$$

where D is a regression window length. For example, if $n = 2$ and $D = 1$ then the first and second order dynamic coefficients can be expressed as the following simple differences

$$\Delta^{(1)}\mathbf{o}^{\mathsf{s}}_t = \frac{1}{2}\left(\mathbf{o}^{\mathsf{s}}_{t+1} - \mathbf{o}^{\mathsf{s}}_{t-1}\right) \tag{2.4}$$

$$\Delta^{(2)}\mathbf{o}^{\mathsf{s}}_t = \frac{1}{2}\left(\Delta^{(1)}\mathbf{o}^{\mathsf{s}}_{t+1} - \Delta^{(1)}\mathbf{o}^{\mathsf{s}}_{t-1}\right) = \frac{1}{4}\mathbf{o}^{\mathsf{s}}_{t+2} - \frac{1}{2}\mathbf{o}^{\mathsf{s}}_t + \frac{1}{4}\mathbf{o}^{\mathsf{s}}_{t-2} \tag{2.5}$$

The first and second order regression coefficients are usually called *delta* and *acceleration* coefficients [265]. The observation vectors produced by the feature extraction stage are usually called *static* coefficients [75]. The complete observation vector $\mathbf{o}_t$ is obtained by appending the delta, acceleration and higher-order regression coefficients to the static coefficients

$$\mathbf{o}_t = \begin{bmatrix}\mathbf{o}^{\mathsf{s}^{\mathsf{T}}}_t & \Delta^{(1)}\mathbf{o}^{\mathsf{s}^{\mathsf{T}}}_t & \ldots & \Delta^{(n)}\mathbf{o}^{\mathsf{s}^{\mathsf{T}}}_t\end{bmatrix}^{\mathsf{T}} \tag{2.6}$$

In the rest of this thesis each observation vector will be assumed to have d elements, i.e., $\mathbf{o}_t \in \mathbb{R}^d$ for all t. The (complete) observation sequence is obtained by

$$\mathbf{O}_{1:T} = \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T \tag{2.7}$$

2.2 Hidden Markov models

For different length speech waveforms, the feature extraction stage yields different length observation sequences. In order to handle variable length sequences, this thesis makes use of sequence models [73]. In this section a popular example is considered - the hidden Markov model (HMM) [12, 188]. The HMM, as adopted in this thesis, is a generative model characterised by [75, 115, 188]

- N, the number of *states* in the HMM. The individual states are denoted by $S_1, S_2, \dots, S_N$, a *hidden* state at time t by q_t and the observed state S_j at time t by q_t^j.

- $\boldsymbol{\pi} = \{\pi_i\}$, the *initial state distribution*, where

$$\pi_i = P(q_1^i) \tag{2.8}$$

  where $P(q_1^i)$ is the probability of observing state S_i at time $t = 1$.

- $\mathbf{A} = \{a_{i,j}\}$, the state *transition probability matrix*, where

$$a_{i,j} = P(q_t^j | q_{t-1}^i), \qquad 1 \leq i, j \leq N \tag{2.9}$$

  where $P(q_t^j | q_{t-1}^i)$ is the probability of a transition from the state S_i occupied at time $t-1$ to the state S_j occupied at time $t+1$. The transition probabilities must satisfy

$$\forall\, i,\ \forall\, j \quad a_{i,j} \geq 0; \qquad \forall\, i \quad \sum_{j=1}^{N} a_{i,j} = 1 \tag{2.10}$$

  to ensure that each row of $\mathbf{A}$ is a valid probability mass function. The *inherent state duration density* has an exponential form

$$p(T|S_j) = a_{j,j}^{T-1}(1 - a_{j,j}) \tag{2.11}$$

which gives the probability of staying T times in state S_j.

- $\mathbf{B} = \{b_j(\cdot)\}$, the set of observation probability measures, where

  $$b_j(\mathbf{o}_t) = p(\mathbf{o}_t|q_t^j) \tag{2.12}$$

  is the state output distribution specifying the *likelihood* of state. A state S_j for which $b_j(\cdot)$ is defined is called *emitting*. In the opposite case it is called *non-emitting*.

The complete set of HMM parameters is denoted by $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$. When more than one HMM is considered then $\boldsymbol{\lambda}$ refers to the set of all HMM parameters. The individual HMM parameters for sentence ω are denoted by $\boldsymbol{\lambda}^{(\omega)}$.

The state output distributions usually adopt probability density functions in the form of Gaussian mixture models (GMM) [115]

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{j,m} \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}) \tag{2.13}$$

where M is the number of mixture components and

$$\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_{j,m}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{o}_t - \boldsymbol{\mu}_{j,m}\right)^{\mathsf{T}} \boldsymbol{\Sigma}_{j,m}^{-1} \left(\mathbf{o}_t - \boldsymbol{\mu}_{j,m}\right)\right) \tag{2.14}$$

is a multivariate normal distribution or Gaussian with *mean vector* $\boldsymbol{\mu}_{j,m}$ and *covariance matrix* $\boldsymbol{\Sigma}_{j,m}$. The individual Gaussians in equation (2.13) can be referred to by

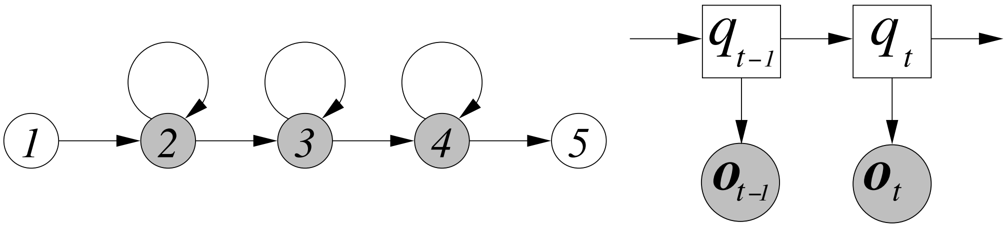
$$b_{j,m}(\mathbf{o}_t) = \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}) \tag{2.15}$$

In order to ensure that the state output distributions are valid probability density functions, the mixture component weights must satisfy

$$\forall\, j,\, \forall\, m \quad c_{j,m} \geq 0; \qquad \forall\, j \quad \sum_{m=1}^{M} c_{j,m} = 1 \tag{2.16}$$

The number of mixture components, M, can be set using simple approaches such as mixture splitting [265] or using more refined approaches such as those described in [34, 69, 143].

Provided the corresponding transition probability is not zero, the HMM can make transition from one state to another. For speech recognition tasks, the HMM usually adopts *strictly left-to-right topology* [10]. The HMM with strictly left-to-right topology in Figure 2.3a does not allow any of the states to be skipped nor transition to be performed in the backward direction.



(a) Strictly left-to-right topology (b) Dynamic Bayesian Network

Figure 2.3: A hidden Markov model. (a) Example of strictly left-to-right topology (blank circle - non-emitting state, shaded circle emitting state, arrow - transition). (b) A dynamic Bayesian network associated with HMM in Figure 2.3b; for simplicity single Gaussian component state output distributions, $M = 1$, are assumed (blank square - discrete hidden state, shaded circle - continuous observation, arrow - statistical dependency).

It is common to visualise assumptions implied by statistical models using dynamic Bayesian networks (DBN) [15, 77]. For the HMM in Figure 2.3a, the corresponding DBN is shown in Figure 2.3b. The DBN shown illustrates two assumptions: *state* and *observation conditional independence* implied by equation (2.9) and (2.12) respectively. The first assumption can be stated as follows: any state is independent of the rest given the previous state [75]. This assumption is reflected in Figure 2.3b by the arrow connecting state q_t only with the previous state q_{t-1}. The second assumption can be stated as follows: any observation is independent of the rest given the state that generated it [75]. This assumption is reflected in Figure 2.3b by the arrow connecting observation $\mathbf{o}_t$ only with the corresponding state q_t.

The likelihood of a particular *state sequence*

$$\mathbf{q}_{1:T} = q_1, q_2, \ldots, q_T \tag{2.17}$$

to produce the observation sequence $\mathbf{O}_{1:T}$ is computed by multiplying the corresponding transition probabilities and likelihoods along time t [75]

$$p(\mathbf{O}_{1:T}, \mathbf{q}_{1:T} | \omega; \boldsymbol{\lambda}) = a_{q_0,q_1} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) a_{q_t,q_{t+1}} \tag{2.18}$$

where $q_0 = S_1$ and $q_{T+1} = S_N$. In practice, only observation sequences are given whilst the underlying state sequences are "hidden", hence the name hidden Markov model. The likelihood assigned by the HMM to observation sequence $\mathbf{O}_{1:T}$ is obtained by summing the likelihood in equation (2.18) over all possible state sequences [188]

$$p(\mathbf{O}_{1:T} | \omega; \boldsymbol{\lambda}) = \sum_{\mathbf{q}_{1:T}} a_{q_0,q_1} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) a_{q_t,q_{t+1}} \tag{2.19}$$

Note that even for small numbers of states and observations, the use of direct summation becomes computationally infeasible due to a large number of possible state sequences [188]. Hence, an algorithm capable of computing equation (2.19) efficiently is required and will be described in Section 2.2.2.

The use of HMM in practical applications requires solutions to the following three standard problems [188]:

- **Optimal state sequence.** Given observation sequence $\mathbf{O}_{1:T}$, sentence ω and HMM parameters $\boldsymbol{\lambda}$, how to find the corresponding state sequence $\mathbf{q}_{1:T}$ optimal in some meaningful sense?

- **Likelihood.** Given observation sequence $\mathbf{O}_{1:T}$, sentence ω and HMM parameters $\boldsymbol{\lambda}$, how to compute the likelihood $p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda})$ efficiently?

- **Parameter estimation.** How to estimate HMM parameters $\boldsymbol{\lambda}$?

Solutions to these problems are considered in Sections 2.2.1, 2.2.2 and 2.2.3.

2.2.1 Viterbi algorithm

Given observation sequence $\mathbf{O}_{1:T}$, sentence ω and HMM parameters $\boldsymbol{\lambda}$, the corresponding state sequence $\mathbf{q}_{1:T}$ is not usually known. There are several possible criteria how one state sequence, optimal in some meaningful sense, can be selected [188]. Among them, a criterion based on maximising the likelihood in equation (2.18) is most commonly used. The state sequence which satisfies this criterion is called the *most likely state sequence* and will be denoted by $\widehat{\mathbf{q}}_{1:T}$. The problem of finding the most likely state sequence can be stated as [188]

$$\widehat{\mathbf{q}}_{1:T} = \arg\max_{\mathbf{q}_{1:T}} \left\{ a_{q_0,q_1} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) a_{q_t,q_{t+1}} \right\} \tag{2.20}$$

A formal technique, known as *Viterbi algorithm* [188], is commonly used to find the most likely state sequence $\widehat{\mathbf{q}}_{1:T}$.

In order to find the most likely state sequence, the Viterbi algorithm introduces the following quantity [188]

$$\phi_j(t) = \max_{\mathbf{q}_{1:t-1}} \left\{ p(\mathbf{O}_{1:t}, \mathbf{q}_{1:t-1}, q_t^j | \omega; \boldsymbol{\lambda}) \right\} \tag{2.21}$$

which is the maximum likelihood of observing the partial observation sequence $\mathbf{O}_{1:t}$ and then being in state S_j at time t [75]. The Viterbi algorithm computes equation (2.21) recursively based on the following recursion [265]

$$\phi_j(t) = \max_{i} \left\{ \phi_i(t-1) a_{i,j} \right\} b_j(\mathbf{o}_t) \tag{2.22}$$

with the initial conditions given by [265]

$$\phi_1(1) = 1, \quad \phi_2(1) = a_{1,2} b_2(\mathbf{o}_1), \quad \ldots \tag{2.23}$$

Upon termination at time $t = T$, the likelihood of $\widehat{\mathbf{q}}_{1:T}$ [265]

$$p(\mathbf{O}_{1:T}, \widehat{\mathbf{q}}_{1:T} | \omega; \boldsymbol{\lambda}) = \max_i \{\phi_i(T) a_{i,N}\} \tag{2.24}$$

In order to retrieve the most likely state sequence, the argument which maximised equation (2.22) is recorded by means of additional quantities

$$\psi_j(t) = \arg\max_i \{\phi_i(t-1) a_{i,j}\} \tag{2.25}$$

The most likely state sequence is retrieved through the following recursion [188]

$$\widehat{q}_t = \psi_{\widehat{q}_{t+1}}(t+1) \tag{2.26}$$

with the initial condition given by

$$\widehat{q}_T = \arg\max_i \{\phi_i(T) a_{i,N}\} \tag{2.27}$$

The computational complexity of the Viterbi algorithm is $\mathcal{O}(N^2 T)$ [49].

2.2.2 Forward-backward algorithm

As noted earlier, the direct computation of likelihood based on equation (2.19) even for small numbers of states and observations is infeasible [188]. However, a formal technique, known as *forward-backward algorithm* [11, 188], is commonly used to efficiently compute the likelihood.

Given observation sequence $\mathbf{O}_{1:T}$, sentence ω and HMM parameters $\boldsymbol{\lambda}$, the forward-backward algorithm introduces the following quantity, known as *forward probability* [188]

$$\alpha_j(t) = p(\mathbf{O}_{1:t}, q_t^j | \omega; \boldsymbol{\lambda}) \tag{2.28}$$

The forward probability $\alpha_j(t)$ is the likelihood of observing the partial observation sequence $\mathbf{O}_{1:t}$ and then being in state S_j and time t. The forward-backward algorithm computes equation (2.28) based on the following recursion [265]

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1) a_{i,j} \right] b_j(\mathbf{o}_t) \tag{2.29}$$

with initial conditions given by [265]

$$\alpha_1(1) = 1, \quad \alpha_2(1) = a_{1,2} b_2(\mathbf{o}_1), \quad \dots \tag{2.30}$$

Upon termination at time $t = T$, the forward probability at the final state S_N is computed by [265]

$$\alpha_N(T) = \sum_{j=2}^{N-1} \alpha_j(T) a_{j,N} \tag{2.31}$$

which is the likelihood of observation sequence $\mathbf{O}_{1:T}$ given class ω and HMM parameters $\boldsymbol{\lambda}$ [265]

$$p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda}) = \alpha_N(T) \tag{2.32}$$

In addition to forward probabilities, the forward-backward algorithm introduces the following quantity, known as *backward probability* [188]

$$\beta_i(t) = p(\mathbf{O}_{t+1:T}|q_t^i, \omega; \boldsymbol{\lambda}) \tag{2.33}$$

The backward probability $\beta_i(t)$ is the likelihood of observing the partial observation sequence $\mathbf{O}_{t+1:T}$ given that at time t the HMM is in state S_i. This probability can be computed based on the following recursion [265]

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{i,j} b_j(\mathbf{o}_{t+1}) \beta_j(t+1) \tag{2.34}$$

with initial condition given by [265]

$$\beta_i(T) = a_{i,N} \tag{2.35}$$

Note that differently to forward probabilities computation is performed starting at time $t = T$ and terminating at time $t = 1$. Upon termination at time $t = 1$, the backward probability at the initial state S_1 is computed by [265]

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1,j} b_j(\mathbf{o}_1) \beta_j(1) \tag{2.36}$$

which is the likelihood of observation sequence $\mathbf{O}_{1:T}$ given class ω and HMM parameters $\boldsymbol{\lambda}$ [265]

$$p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda}) = \beta_1(1) \tag{2.37}$$

The computational complexity of forward-backward algorithm is $\mathcal{O}(N^2T)$ [49]. Compared to equation (2.19), the likelihood defined by equation (2.32) or (2.37) requires orders of magnitude less computation [188].

The forward-backward algorithm can be also used to compute a posterior probability of occupying state S_j at time t [188]

$$\gamma_j(t) = P(q_t^j|\mathbf{O}_{1:T}, \omega; \boldsymbol{\lambda}) \tag{2.38}$$

The posterior probability $\gamma_j(t)$ is computed by means of forward probability $\alpha_j(t)$, backward probability $\beta_j(t)$ and likelihood $p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda})$ by [188]

$$\gamma_j(t) = \frac{\alpha_j(t)\beta_j(t)}{p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda})} \tag{2.39}$$

In the following, these posterior probabilities are called *state occupancies*. A posterior probability of occupying state S_j and component m can be obtained by [265]

$$\gamma_{j,m}(t) = \frac{\alpha_j(t)\frac{c_{j,m}b_{j,m}(\mathbf{o}_t)}{b_j(\mathbf{o}_t)}\beta_j(t)}{p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda})} \tag{2.40}$$

In the following, these posterior probabilities are called *state-component occupancies*. In addition to occupancies, a posterior probability of occupying state S_i at time t and state S_j at time $t+1$ can be obtained by

$$\zeta_{i,j}(t) = \frac{\alpha_i(t)a_{i,j}b_j(\mathbf{o}_{t+1})\beta_j(t+1)}{p(\mathbf{O}_{1:T}|\omega; \boldsymbol{\lambda})} \tag{2.41}$$

In the following, these posterior probabilities are called *state-state occupancies*.

As will be shown in Sections 2.2.3 and 2.7, all these occupancies play a fundamental role in estimating HMM parameters.

2.2.3 Maximum likelihood estimation

As noted earlier, the use of HMMs in practical applications requires knowing how to estimate HMM parameters. The observation sequences used to estimate HMM parameters are called *training sequences* [188]. This thesis is concerned only with a *supervised training*, where reference transcriptions are provided for all training sequences, whilst alternative settings, such as lightly supervised and unsupervised training [75], are not examined. Thus, the training data $\mathcal{D}$ is

$$\mathcal{D} = \left\{ \left\{ \mathbf{O}^{(1)}_{1:T_1}, \omega_1 \right\}, \dots, \left\{ \mathbf{O}^{(r)}_{1:T_r}, \omega_r \right\}, \dots, \left\{ \mathbf{O}^{(R)}_{1:T_R}, \omega_R \right\} \right\} \tag{2.42}$$

where ω_r is the r-th reference transcription and $\mathbf{O}^{(r)}_{1:T_r}$ is the r-th observation sequence consisting of T_r observations. There are several criteria how one set of HMM parameters, optimal in some meaningful sense, can be selected [8, 12]. Among them, a criterion based on maximising the likelihood in equation (2.18) will be discussed in the rest of this section.

The maximum likelihood (ML) criterion [12] aims to maximise the likelihood that the HMM generates the training data. The ML objective function may be expressed as [75]

$$\mathcal{F}_{\texttt{ml}}(\boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \log(p(\mathbf{O}^{(r)}_{1:T_r} | \omega_r; \boldsymbol{\lambda})) \tag{2.43}$$

There is no known way to analytically solve for the HMM parameters which maximise equation (2.43) [188]. In order to address this issue, a number of alternative approaches, such as the standard multi-dimension optimisation [171] and *Baum-Welch algorithm* [12], have been considered [188]. The use of standard multi-dimension optimisation techniques [171] may result in slow training times as the dimensionality of this problem is usually large [61]. In contrast, the Baum-Welch algorithm has been empirically found to converge in few iterations and is commonly adopted [75, 188, 265].

The Baum-Welch algorithm - an instance of *expectation-maximisation* (EM) technique [45] - is an iterative procedure where, given the current set of HMM parameters $\boldsymbol{\lambda}$, a new set of HMM parameters $\widehat{\boldsymbol{\lambda}}$ is found such that [188]

$$\mathcal{F}_{\mathtt{ml}}(\widehat{\boldsymbol{\lambda}};\mathcal{D}) \geq \mathcal{F}_{\mathtt{ml}}(\boldsymbol{\lambda};\mathcal{D}) \tag{2.44}$$

is guaranteed to hold. In order to find $\widehat{\boldsymbol{\lambda}}$, an *auxiliary function* $\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D})$ is introduced [188]

$$\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{q}_{1:T_r}^{(r)}} P(\mathbf{q}_{1:T_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)},\omega_r;\boldsymbol{\lambda})\log\left(p(\mathbf{O}_{1:T_r}^{(r)},\mathbf{q}_{1:T_r}^{(r)}|\omega_r;\widehat{\boldsymbol{\lambda}})\right) \tag{2.45}$$

where $\mathbf{q}_{1:T_r}^{(r)}$ is a *state-component sequence*, $P(\mathbf{q}_{1:T_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)},\omega_r;\boldsymbol{\lambda})$ is a posterior probability of $\mathbf{q}_{1:T_r}^{(r)}$,

$$p(\mathbf{O}_{1:T_r}^{(r)},\mathbf{q}_{1:T_r}^{(r)}|\omega_r;\widehat{\boldsymbol{\lambda}}) = \widehat{a}_{q_0^r,q_1^r}\prod_{t=1}^{T_r}\widehat{c}_{q_t^r}\mathcal{N}(\mathbf{o}_t^{(r)};\widehat{\boldsymbol{\mu}}_{q_t^r},\widehat{\boldsymbol{\Sigma}}_{q_t^r})\widehat{a}_{q_t^r,q_{t+1}^r} \tag{2.46}$$

is the likelihood of $\mathbf{q}_{1:T_r}^{(r)}$ to produce $\mathbf{O}_{1:T_r}^{(r)}$ based on the new set of HMM parameters. Although $\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D})$ contains the summation over state-component sequences, it can be re-written in terms of state-component $\gamma_{j,m}^{(r)}(t)$ and state-state $\zeta_{i,j}^{(r)}(t)$ occupancies (Section 2.2.2) in the following way [135, 142]

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) \quad = \quad & \frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{\{j,m\}}\gamma_{j,m}^{(r)}(t)\left(\log(\widehat{c}_{j,m}) + \log(\mathcal{N}(\mathbf{o}_t^{(r)};\widehat{\boldsymbol{\mu}}_{j,m},\widehat{\boldsymbol{\Sigma}}_{j,m}))\right) + \\ & \frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{\{i,j\}}\zeta_{i,j}^{(r)}(t)\log(\widehat{a}_{i,j})\end{aligned} \tag{2.47}$$

Not only it is possible to analytically solve for the HMM parameters which maximise $\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D})$ but it was also proved that maximising $\mathcal{Q}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D})$ also increases $\mathcal{F}_{\mathtt{ml}}(\boldsymbol{\lambda};\mathcal{D})$ [12, 113, 142]. The derivation of closed-form expressions for the new set of HMM parameters has been extensively covered in the literature [113, 142].

As an illustrative example, consider estimating HMM mean vectors and covariance matrices. Given the training data, the forward-backward algorithm described in Section 2.2.2 is applied to each training observation sequence to accumulate the following statistics

$$\gamma_{j,m} = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_{j,m}^{(r)}(t) \tag{2.48}$$

$$\boldsymbol{\theta}_{j,m} = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_{j,m}^{(r)}(t)\mathbf{o}_t^{(r)} \tag{2.49}$$

$$\boldsymbol{\Theta}_{j,m} = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_{j,m}^{(r)}(t)\mathbf{o}_t^{(r)}\mathbf{o}_t^{(r)\mathsf{T}} \tag{2.50}$$

where $\gamma_{j,m}^{(r)}(t)$ is the state-component occupancy in the r-th training sequence. This statistics is usually called the HMM *occupancy*, *mean* and *covariance statistics* respectively [265]. This computation corresponds to the *expectation step* (E) of the EM technique [75]. Given the statistics, the new HMM mean vector and covariance matrix are obtained by [188]

$$\widehat{\boldsymbol{\mu}}_{j,m} = \frac{\boldsymbol{\theta}_{j,m}}{\gamma_{j,m}} \tag{2.51}$$

$$\widehat{\boldsymbol{\Sigma}}_{j,m} = \frac{\boldsymbol{\Theta}_{j,m}}{\gamma_{j,m}} - \widehat{\boldsymbol{\mu}}_{j,m}\widehat{\boldsymbol{\mu}}_{j,m}^{\mathsf{T}} \tag{2.52}$$

These update rules correspond to the *maximisation step* (M) of EM technique [75]. The statistics used by the algorithm can be accumulated for one or more training sequences independently of the rest, which allows efficient parallel implementation [265]. The algorithm is usually applied few times, each time replacing $\boldsymbol{\lambda}$ with $\widehat{\boldsymbol{\lambda}}$ and repeating the E and the M step to obtain new $\widehat{\boldsymbol{\lambda}}$ [172].[1]

2.3 Composite sentence modelling

The HMM described in Section 2.2 was presented as the whole-sentence statistical model $p(\mathbf{O}_{1:T}|\omega;\boldsymbol{\lambda})$ which associates parameters $\boldsymbol{\lambda}^{(\omega)}$ with sentence ω. As

[1] In practice, a typical built of HMM-based acoustic models, such as the one described in HTK manual [265], is a multi-stage procedure, where each stage involves 3-4 ML iterations.

the number of possible sentences increases such approach quickly becomes impractical. In order to address this issue, a *structure* can be introduced into the statistical model, where all sentences are broken down into sub-sentence units and modelled by combining units into a composite sentence statistical model [73, 238]. The next Section 2.3.1 discusses selection of appropriate sub-sentence units. The following Section 2.3.2 discusses how HMM-based composite sentence models can be constructed for a given selection of sub-sentence units.

2.3.1 Unit selection

The set of sub-sentence units is usually called *vocabulary* [75]. The simplest choose of sub-sentence units are words [23, 188, 212]. An example decomposition of sentence ω into L-length word sequence $\mathbf{w}_{1:L}$ is

$$\begin{aligned} \omega &= \{\texttt{<s> the dog chased the cat </s>}\} \\ \mathbf{w}_{1:7} &= \{\texttt{<s>}, \texttt{the}, \texttt{dog}, \texttt{chased}, \texttt{the}, \texttt{cat}, \texttt{</s>}\} \end{aligned} \tag{2.53}$$

where `<s>` and `</s>` mark the beginning and end of the sentence.

As the size of vocabulary increases it becomes harder to obtain good coverage of words by the training data so for large vocabulary tasks it is more common to adopt sub-word units such as *phones* [75, 188, 264]. The set of context-independent phones are usually called *monophones* [75]. In order to decompose words into phone units, a pronunciation dictionary is required. The pronunciation dictionary specifies one or more possible phone sequences together with the corresponding pronunciation probabilities for each word. An excerpt in Table 2.1, where the last two entries illustrate the use of multiple pronunciations, shows one typically used format [265].

The less complex and fewer units are selected, the more variable they are [188]. For instance, the same phone may be realised differently due to co-articulation, stress and other factors [144]. Acoustic models that capture this variability are expected to provide a more consistent and accurate representation of speech [174]. One approach to capture that variability in a systematic way is to make use of *context-dependent* phone units [75, 174]. A simple example of context-dependent phone unit is a *triphone*, which takes both left and right context into account.

Word	Pronunciation probability	Phone sequence
`<s>`	1.0	`sil`
`</s>`	1.0	`sil`
`cat`	1.0	`/k/, /ae/, /t/`
`chased`	1.0	`/ch/, /ey/, /s/, /t/`
`dog`	1.0	`/d/, /ao/, /g/`
`the`	0.7	`/dh/, /iy/`
`the`	0.3	`/dh/, /ah/`

Table 2.1: An excerpt from pronunciation dictionary

For instance, `/t/` preceded by `/s/` and followed by `/dh/` is the triphone `s-t+dh`. The context-dependence is known to spread across word boundaries which is essential for capturing many important phonological processes [75]. The resulting set of context-dependent phone units is called *cross-word*. Figure 2.4 illustrates a complete decomposition of sentence into context-dependent phones, where the most likely according to Table 2.1 pronunciations have been adopted.

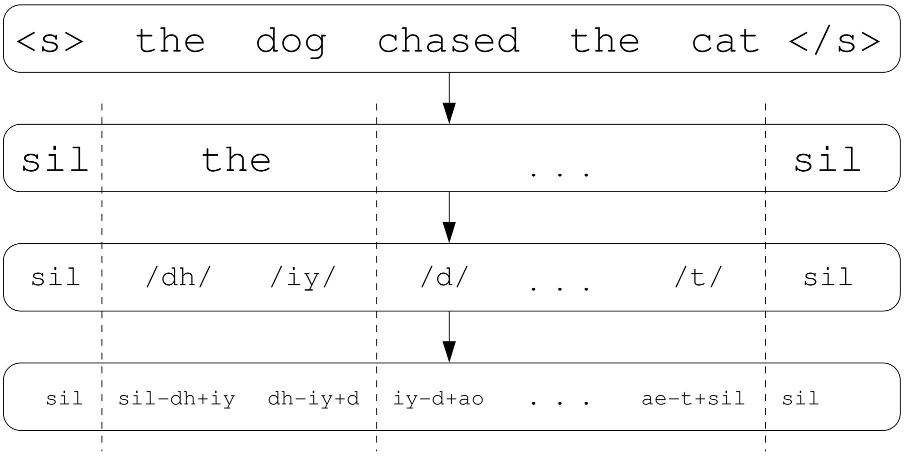
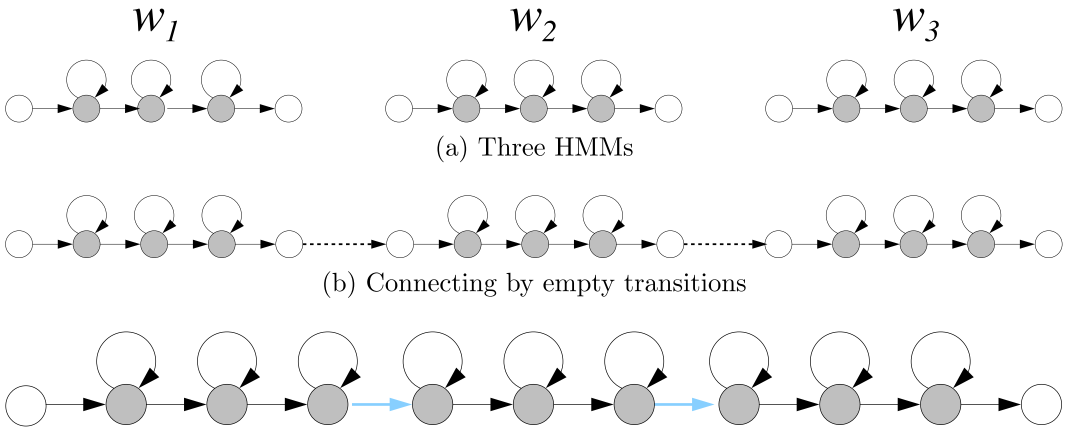


Figure 2.4: An example of decomposing sentence into words, words into phones and converting phones into context-dependent phones

In general, the more context is taken into account, the more accurate modelling of speech is expected [9]. However, as the amount of context taken into account increases, it becomes harder to obtain good coverage in the training data [174]. For instance, the use of cross-word set typically yields a large number of units with few examples [266]. Furthermore, some units required during testing may have not been seen in the training data at all. This problem of *unseen units* becomes the more severe, the more context is taken into account [174]. A standard approach to address these issues will be discussed in Section 2.4.

2.3.2 Composite HMMs

Given the set of sub-sentence units, rather than associating the statistical model parameters with individual sentences these now can be associated with the individual sub-sentence units. The composite sentence statistical model is then formed by combining multiple units together. When the HMM is used as the statistical model then the composite sentence statistical model is called a *composite HMM* [172]. The composite HMM may be constructed by ”gluing” the HMMs associated with the units together in two steps as illustrated by Figure 2.5 where these units are words. The first step joins the individual HMMs by means of empty transitions as shown by the dashed arrows in Figure 2.5b. The second step subsumes exit, empty and entry transitions of the HMMs into intra-model transitions as shown by the light arrows in Figure 2.5c. If each unit in $\mathbf{w}_{1:L}$ is modelled by an N-state HMM then the composite HMM for word sequence $\mathbf{w}_{1:L}$



(a) Three HMMs

(b) Connecting by empty transitions

(c) Merging into single composite HMM

Figure 2.5: An example of converting a sequence of three HMMs into single composite HMM.

is $L(N-2)+2$-state HMM with parameters $\boldsymbol{\lambda}^{(\omega)} = \{\boldsymbol{\lambda}^{(w_1)}, \dots, \boldsymbol{\lambda}^{(w_L)}\}$ such that

$$p(\mathbf{O}_{1:T}|\omega;\boldsymbol{\lambda}) = p(\mathbf{O}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda}) = \sum_{\mathbf{q}_{1:T}} a_{q_0,q_1} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) a_{q_t,q_{t+1}} \tag{2.54}$$

where $\mathbf{q}_{1:T}$ is a state sequence in the composite HMM. The use of units other than words can be handled in a similar way [172]. The composite HMM thus constructed can adopt the Viterbi algorithm (Section 2.2.1), the forward-backward algorithm (Section 2.2.2) and ML parameter estimation (Section 2.2.3) providing the solutions to the three standard HMM problems described in Section 2.2.

2.4 Phonetic decision trees

As noted earlier, the use of context-dependent phone units introduces two conflicting requirements: for accurate modelling of speech the amount of both context and training data per unit should be as large as possible. In order to address the data sparsity problem, the complete set of *logical* units can be clustered into a reduced set of *physical* units [75]. The standard approach, is based on a top-down phonetic decision tree clustering [174, 266].

A phonetic decision tree is a binary tree in which a question is attached to each node [266]. For instance, in Figure 2.6, the question ”Is the phone on the left of the current phone a vowel?” is attached to the root node. If the answer is ”yes” then the question attached to the left child node is asked, otherwise the question

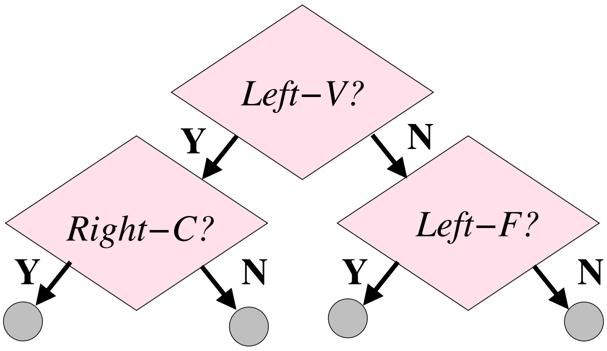


Figure 2.6: An example of phonetic decision tree

attached to the right child node is asked. In the end, a terminal node is reached where no questions are asked as illustrated by shaded circles in Figure 2.6.

There are options how clustering of logical units into physical units can be performed [174, 266]. One options is to construct one phonetic decision tree for each phone. For instance, the phonetic decision tree in Figure 2.6 will partition its phones into four subsets as indicated by the four terminal nodes. The phones in each subset are tied to form a single *physical model*. This is an example of *model-level tying*. Another option is to construct one tree for each emitting state of each phone to cluster all of the corresponding emitting states of all of the associated context-dependent phone units [266]. For instance, the phonetic decision tree in Figure 2.6 will partition its states into four subsets as indicated by the four terminal nodes. The states in each subset are tied to form a single *physical state*. The physical models are constructed by combining the decision trees associated with the corresponding emitting states. This is an example of *state-level tying*.

The clustering of logical to physical units typically operates at the state-level rather than model-level since it allows a larger set of physical units to be robustly estimated by combining state-level decision trees to yield physical models [75]. When several decision trees are combined then the resulting acoustic model is known to be subject to a *tree-intersect effect* [268]. Figure 2.7 shows an example of acoustic model combining two phonetic decision trees each providing 4 physical states. Although the number of physical states is 8, the number of physical models that can be constructed is 16. The larger the number of physical states, the larger the number of physical models. Thus, the effective number of physical model can become very large, which may result in robustness issues when training the acoustic model [191].

The phonetic decision tree construction is performed using sequential optimisation procedure [174, 265] and can be summarised in the following three steps [174, 266]:

1. **Splitting.** During construction, the decision which terminal node to split and which question to select is based on the criterion maximising the likelihood of training data whilst ensuring that there is sufficient training data available to each terminal node given the current set of physical states.

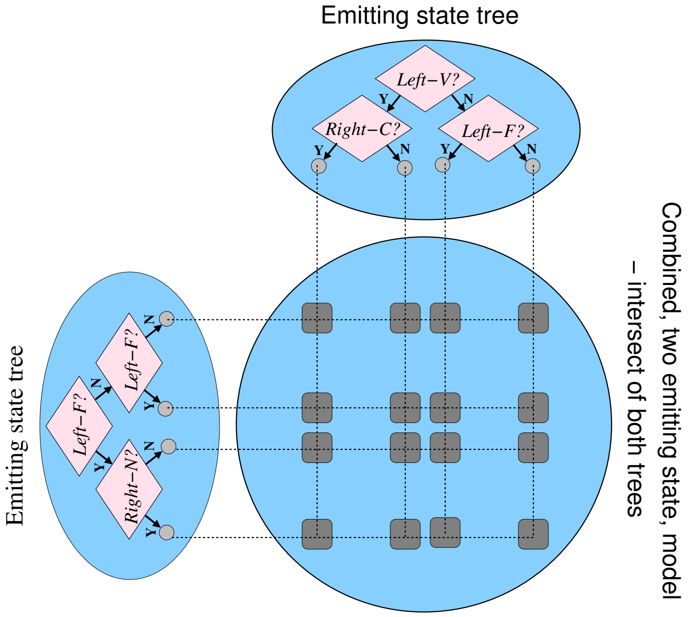


Figure 2.7: An example of tree-intersect acoustic model. The physical states are shown by shaded circles. The physical models are shown by shaded squares

2. **Termination.** The construction stops once one of the following two conditions holds: (a) the increase in likelihood from splitting every terminal node is below a threshold, (b) the amount of training data available to either the left or the right child potentially to be created is not sufficient.

3. **Projection.** Once the phonetic decision trees have been constructed, the unseen as well as the seen units are synthesised by finding the appropriate terminal nodes for contexts of that unit and then using the physical states associated with those terminal nodes to construct the unit. Among units sharing the same set of physical states one is selected to provide a context-dependent phone label. The remaining units are mapped to this label.

The theoretical framework behind phonetic decision tree construction can be summarised as follows [174, 266]. Let $\mathbf{S}$ be a set of states associated with a terminal node and let $\ell(\mathbf{S})$ be the log-likelihood of $\mathbf{S}$ generating the training data $\mathbf{O}^{(1)}_{1:T_1}, \ldots, \mathbf{O}^{(R)}_{1:T_R}$ under the assumption that all states $S \in \mathbf{S}$ are tied (one physical state) and that transition probabilities can be ignored. Assuming that the state output distributions are Gaussians with the common mean $\boldsymbol{\mu}(\mathbf{S})$ and variance $\mathbf{\Sigma}(\mathbf{S})$, tying does not affect observation/state alignment, the following approximation for $\ell(\mathbf{S})$ is used [266]

$$\ell(\mathbf{S}) = -\frac{1}{2}\left(\log\left((2\pi)^d|\mathbf{\Sigma}(\mathbf{S})|\right) + d\right)\gamma(\mathbf{S}) \tag{2.55}$$

where d is the dimensionality of observation vectors and $\gamma(\mathbf{S})$ is the total occupancy of the tied states

$$\gamma(\mathbf{S}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{S\in\mathbf{S}}\gamma_S^{(r)}(t) \tag{2.56}$$

which is adopted as the measure of training data sufficiency. The posterior probability of occupying state S at time t given the r-th observation sequence, $\gamma_S^{(r)}(t)$, can be computed using the forward-backward algorithm (Section 2.2.2). For the terminal node to be split $\gamma(\mathbf{S})$ is required to exceed a threshold. Let $\mathbf{S}_{\mathtt{y}}(v)$ be the subset of states from $\mathbf{S}$ which context-dependent labels answer "yes" to the question v and $\mathbf{S}_{\mathtt{n}}(v)$ contains the remaining. The decision to split the terminal node is made if

$$\Delta\ell_v(\mathbf{S}) = \ell(\mathbf{S}_{\mathtt{y}}(v)) + \ell(\mathbf{S}_{\mathtt{n}}(v)) - \ell(\mathbf{S}) \tag{2.57}$$

and, $\gamma(\mathbf{S}_{\mathtt{y}}(v))$ and $\gamma(\mathbf{S}_{\mathtt{n}}(v))$ for some v exceed their associated thresholds. If they do, the question maximising equation (2.57) splits the terminal node. If none of the terminal nodes can be split then the construction stops. One important aspect of the approximation for $\ell(\mathbf{S})$ in equation (2.55) is that it depends *only* on the tied variance $\mathbf{\Sigma}(\mathbf{S})$ and the total occupancy of the tied states $\gamma(\mathbf{S})$. In addition, under the assumptions behind it, $\gamma(\mathbf{S})$ can be computed only once thus avoiding making any reference to the training data which yields computationally efficient phonetic decision tree construction procedure.

2.5 Language modelling

In addition to acoustic model, the Bayes' rule in equation (1.1) requires *language model* to provide the prior probability of sentences $P(\omega)$. As the number of possible sentences increases it becomes complicated to model individual sentences.

In order to overcome this issue, a *structure* can be introduced into the language model [73]. The standard approach is based on decomposing sentence ω into word sequence $\mathbf{w}_{1:L}$ [31, 73, 75, 265]. The prior probability then can be expressed using chain rule as follows [31]

$$P(\omega) = P(\mathbf{w}_{1:L}) = \prod_{i=1}^{L} P(w_i|w_{i-1},\dots,w_1) \tag{2.58}$$

where $P(w_i|w_{i-1},\dots,w_1)$ is the probability of word w_i given history of all previous words. Truncating history to $n-1$ most recent words yields n-gram model [31]

$$P(\omega) = P(\mathbf{w}_{1:L}) = \prod_{i=1}^{L} P(w_i|w_{i-1},\dots,w_{i-n+1}) \tag{2.59}$$

The n-tuple of words, $\{w_{i-n+1},\dots,w_{i-1},w_i\}$, in equation (2.59) is known as the n-gram, hence the name *n-gram model* [265]. In practice, n is typically set in the range of two to four [75]. The assumptions implied by the n-gram model can be visualised using DBN [15, 77]. Figure 2.8 shows the DBN associated with bigram models.

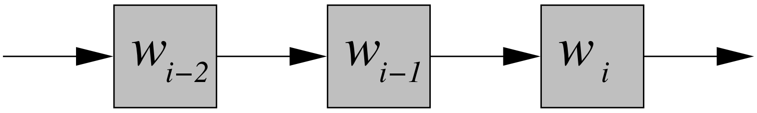


Figure 2.8: Dynamic Bayesian network for bigram language model

The conditional probabilities $P(w_i|w_{i-1},\dots,w_{i-n+1})$ in equation (2.59) are estimated from training texts [75]. Let $C(w_{i-n+1},\dots,w_{i-1},w_i)$ be the number of times the underlying n-gram occurs in training texts. A maximum likelihood (ML) estimate is then given by [31, 75]

$$P(w_i|w_{i-1},\dots,w_{i-n+1}) = \frac{C(w_{i-n+1},\dots,w_{i-1},w_i)}{C(w_{i-n+1},\dots,w_{i-1})} \tag{2.60}$$

By increasing the order n, the accuracy of approximation in equation (2.59) may be expected to improve [159, 212]. However, it is complicated to ensure sufficient coverage in training texts and handle unseen n-grams - the *data sparsity* *problem* [75, 159]. In order to address this issue, a combination of discounting and backing-off is commonly used [75]. For instance, the Katz smoothing scheme [31, 75, 106, 119] sets conditional probabilities by

$$P(w_i|w_{i-1},\dots,w_{i-n+1}) = \tag{2.61}$$
$$\begin{cases} D\dfrac{C(w_{i-n+1},\dots,w_{i-1},w_i)}{C(w_{i-n+1},\dots,w_{i-2},w_{i-1})}, & \text{if} \quad 0 < C(w_{i-n+1},\dots,w_{i-1},w_i) \le C^{\mathtt{min}} \\ \dfrac{C(w_{i-n+1},\dots,w_{i-1},w_i)}{C(w_{i-n+1},\dots,w_{i-2},w_{i-1})}, & \text{if} \quad C(w_{i-n+1},\dots,w_{i-1},w_i) > C^{\mathtt{min}} \\ \dfrac{P(w_i|w_{i-1},\dots,w_{i-n+2})}{Z(w_{i-n+1},\dots,w_{i-2},w_{i-1})}, & \text{otherwise} \end{cases}$$

where D is a discounting coefficient for n-grams observed less than $C^{\mathtt{min}}$ times in training texts and $Z(w_{i-n+1},\dots,w_{i-2},w_{i-1})$ is a normalisation constant. The goal of discounting is to reserve probability mass for the unseen n-grams [159]. There are several options how the discounting coefficient D can be set [31]. For instance, the Good-Turing scheme [79] discounts n-grams occurring exactly c times by [75, 265]

$$D = \frac{(c+1)C_{c+1}}{c\,C_c} \tag{2.62}$$

where C_c is the number of n-grams occurring exactly c times in the training texts. For unseen n-grams, an estimate is obtained from the third case in equation (2.61), which uses the estimate of conditional probability associated with the $n-1$-gram scaled by the normalisation constant. The use of normalisation constants ensures that equation (2.61) yields a valid probability mass function.

In addition to n-gram models, a range of other language models have been investigated, such as class n-gram models [24, 159], maximum entropy language models [199], neural network language models [13], to name a few.

2.6 Decoding and lattice generation

A hypothesised word sequence encoded into given observation sequence can be found by searching all possible hidden state sequences arising from all possible word sequences for the sentence which most likely have generated the observation sequence [75]. If all possible word sequences can be compactly encoded into a single composite HMM (Section 2.3) then an efficient solution to the decoding problem can be found by means of the Viterbi algorithm (Section 2.2). In practice, the use of Viterbi algorithm for decoding becomes unmanageably complex due to the topology, the n-gram language model constraints, the use of cross-word context-dependent units and the size of memory required to hold the composite HMM [75]. A number of approaches have been proposed to address these issues such as dynamic decoding [174, 177], stack decoding [105], static decoding based on weighted finite-state transducer (WFST) technology [158]. However, a comprehensive description of this topic is out of the scope of this thesis, for more information consider [75, 174] and reference therein.

2.6.1 N-best lists

Although the primary task of decoder consist of finding the hypothesised word sequence which most likely have generated the observation sequence, it is also usually possible to output N most likely candidates or the N-*best list* [265]. An example of N-best list is given by

$$\underbrace{\begin{array}{ccc} 1 & 50 & \texttt{sil} \\ 51 & 130 & \texttt{the} \\ 131 & 260 & \texttt{dog} \\ \vdots & & \\ 531 & 560 & \texttt{sil} \end{array}}_{\text{1-best}} \;,\ldots,\; \underbrace{\begin{array}{ccc} 1 & 50 & \texttt{sil} \\ 51 & 110 & \texttt{the} \\ 111 & 150 & \texttt{cat} \\ \vdots & & \\ 311 & 560 & \texttt{sil} \end{array}}_{N-\text{best}} \tag{2.63}$$

where each candidate is shown in the three column format: first observation, last observation, word [265]. In addition, acoustic and language model scores may be specified. The use of N-best lists is useful as it allows multiple passes over the observation sequence without the computational expense of repeatedly solving the decoding problem from scratch [75]. For instance, the first pass can be performed using less complex acoustic and language models to produce an N-best list. The N-best list then can be re-scored candidate-by-candidate using more complex models.

2.6.2 Lattices

As the number of candidates increases, the use of N-best lists becomes computationally and memory inefficient. In order to store N-best lists in a compact and efficient manner, the use of *word lattices* can be adopted [174, 230, 265]. A word lattice consists of a set of nodes representing points in time and a set of spanning arcs representing word hypotheses [75]. Figure 2.9 shows an example of word lattice encoding the word sequence in equation (2.53) as well as several alternative candidates. In addition to words, each arc can also carry additional information such as acoustic, pronunciation and language model scores.

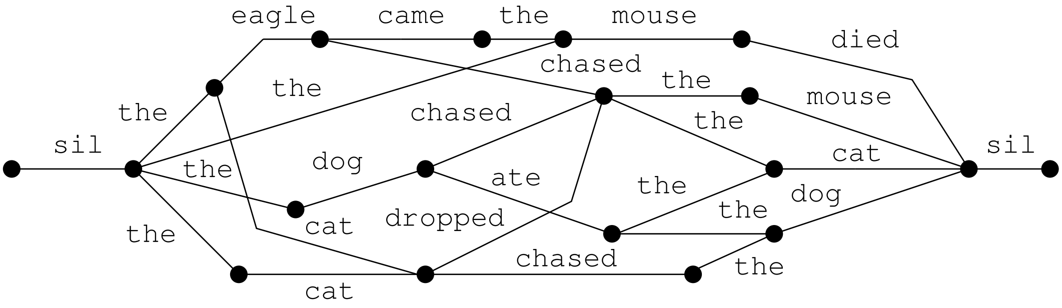


Figure 2.9: An example of word lattice.

Lattices have a wide-spread use in speech recognition [75]. For instance, they can be converted into an efficient representation called *confusion network* [52, 75, 150]. The confusion network has the important property that for every path through the original word lattice, there is a corresponding path through the confusion network [75]. Figure 2.10 shows an example of confusion network for the word lattice in Figure 2.9. In the confusion network, each set of parallel

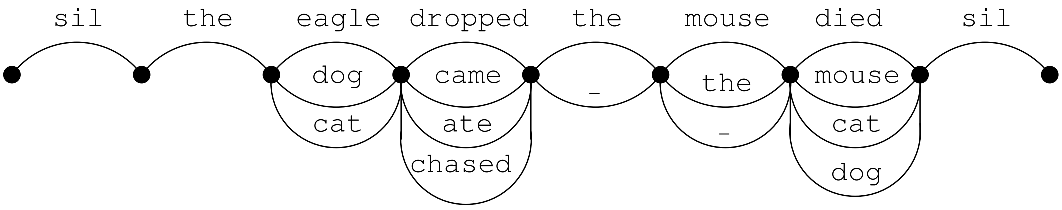


Figure 2.10: An example of confusion network.

arcs represents word hypotheses which, unlike in word lattices, do not necessarily exactly overlap in time. Nevertheless, it is assumed that the amount of overlap is sufficient to treat the set of parallel arcs as the set of competing hypotheses [75]. In addition to word label, each arc in the confusion network has a start and end time information, and word posterior probability [52]. Confusion networks have been commonly used for minimum word-error decoding [26, 218], which attempts to hypothesise word sequence minimising the word-error rate rather than maximising the posterior probability according to Bayes' decision rule in equation (1.1), to provide confidence scores about reliability of decoded hypotheses [88], for merging the outputs of different decoders [52, 54] and in combination with discriminative classifiers [247].

Lattices can also be converted into phone-marked lattices [265]. A *phone-marked* lattice is an extension to word lattice where each word arc is split into *phone arcs* corresponding to the underlying sequence of phones. Each phone arc can contain acoustic model scores, such as the HMM likelihood associated with the phone. Figure 2.11 gives an example of phone-marked lattice illustrating phone-level acoustic model scores and word-level pronunciation and language model scores. The phone arcs are connected by means of phone arc transitions

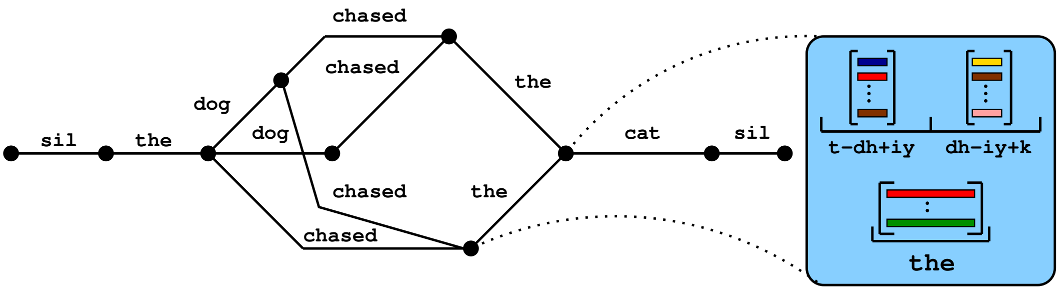


Figure 2.11: An example of phone-marked lattice showing phone-level acoustic model features and word-level pronunciation and language model features

which have associated costs [265]. For instance, the cost of transiting from the last phone arc of one word arc into the first phone arc of another word arc is set equal to the product of language and pronunciation model scores attached to the latter word arc. The total cost associated with traversing the phone-marked lattice can be obtained by summing the costs associated with individual phone arc sequences. The total cost of phone-marked lattice $\mathbb{L}$, where acoustic model scores are HMM likelihoods, may be expressed as [265]

$$[[\mathbb{L}]] = \sum_{\mathbf{a}\in\mathbb{L}} P(a_1|a_0) \prod_{s=1}^{|\mathbf{a}|} p(\mathbf{O}_{\{a_s\}}|a_s^{\mathtt{i}};\boldsymbol{\lambda}) P(a_{s+1}|a_s) \tag{2.64}$$

where $\mathbf{a}$ is a phone arc sequence, a_s is the s-th phone arc in $\mathbf{a}$, $a_s^{\mathtt{i}}$ is an identity of a_s such as `t-dh+iy` in Figure 2.11, $\mathbf{O}_{\{a_s\}}$ is an observation sub-sequence spanned by a_s where $\{a_s\}$ denotes the associated range of observation vector indices, $p(\mathbf{O}_{\{a_s\}}|a_s^{\mathtt{i}};\boldsymbol{\lambda})$ is the HMM likelihood and $P(a_{s+1}|a_s)$ is a phone arc transition probability. For convenience, the first a_0 and last $a_{|\mathbf{a}|+1}$ arc is mapped to the common sentence beginning $a_{\mathcal{I}}$ and end $a_{\mathcal{F}}$ phone arc respectively. The total cost or *lattice weight* (also known as the lattice/acceptor weight in WFST terminology [156, 158]) can be efficiently computed using *lattice forward-backward algorithm* [184, 240]. Similar to the standard forward-backward algorithm (Section 2.2.2), the lattice forward-backward algorithm introduces two recursions [184]

$$\alpha_a = p(\mathbf{O}_{\{a\}}|a^{\mathtt{i}};\boldsymbol{\lambda}) \sum_{a' \text{ preceding } a} \alpha_{a'} P(a|a') \tag{2.65}$$

$$\beta_a = \sum_{a' \text{ following } a} p(\mathbf{O}_{\{a'\}}|a'^{\mathtt{i}};\boldsymbol{\lambda}) \beta_{a'} P(a'|a) \tag{2.66}$$

where α_a and β_a are the forward and backward probability on phone arc a. The posterior probability to occupy phone arc a or *arc occupancy* is given by [184]

$$\gamma_a = \frac{\alpha_a \beta_a}{[[\mathbb{L}]]} \tag{2.67}$$

The lattice weight $[[\mathbb{L}]]$ can be obtained by

$$[[\mathbb{L}]] = \alpha_{a_{\mathcal{F}}} = \beta_{a_{\mathcal{I}}} \tag{2.68}$$

2.6.3 Character, word and sentence error rates

The accuracy of hypothesised word sequences is commonly assessed by comparing them against known reference transcriptions [265]. When the accuracy of correctly recognising sub-sentence units such as words or characters is of interest then comparison is performed by aligning the hypothesised word sequences against the reference transcriptions using dynamic alignment algorithms [157, 265]. The number of substitution, deletion and insertion errors is then computed by comparing the aligned sequences. Summing the numbers of errors and dividing by the number of sub-sentence units in the reference transcriptions yields in percentage points (×100%) word- (WER) or character-error rate (CER). When accuracy of correctly recognising entire sentences is of interest then the number of incorrectly recognised sentences in percentage points, called sentence-error rate (SER), can be computed.

2.7 Discriminative parameter estimation

In order to provide solution to the HMM estimation problem, the use of ML criterion was discussed in Section 2.2.3. The ML criterion aims to estimate HMM parameters so that the likelihood that the HMM generates the training data is maximised [75]. For ML to be an optimal parameter estimation criterion, a number of conditions would need to be met, such as model correctness and training data sufficiency [22, 165]. The use of conditional independence assumptions in the HMM (Section 2.2) and finite amounts of training data are believed to violate those conditions [75].

In order to address this issue, a range of alternative, *discriminative criteria* have been developed. The discriminative criteria aim to estimate HMM parameters so that hypotheses generated by decoder on the training data more closely ”match” the reference transcriptions whilst generalising to unseen test data [75]. An overview of discriminative criteria is given in Section 2.7.1.

Compared to ML, where there exists efficient Baum-Welch algorithm, the use of discriminative criteria is complicated due to more complex implementation required and tendency of these criteria to *over-train*, which may lead to poor generalisation to the test data [75, 184, 186]. A range of solutions that have been developed to address these issues is discussed in Section 2.7.2.

2.7.1 Discriminative criteria

As noted earlier, a range of discriminative criteria can be adopted as an alternative to ML estimation of HMM parameters. Examples discussed in this section include maximum mutual information (Section 2.7.1.1), minimum classification error (Section 2.7.1.2), minimum Bayes' risk (Section 2.7.1.3), margin-based (Section 2.7.1.4) and perceptron (Section 2.7.1.5) criteria. All these discriminative criteria can be expressed in terms of posterior probabilities associated with word sequences [64, 73, 75]. Using Bayes' rule, the posterior of word sequence $\mathbf{w}_{1:L}$ given observation sequence $\mathbf{O}_{1:T}$ and HMM parameters $\boldsymbol{\lambda}$ can be expressed as

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T};\boldsymbol{\lambda}) = \frac{p(\mathbf{O}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})P(\mathbf{w}_{1:L})}{\sum_{\mathbf{w}} p(\mathbf{O}_{1:T}|\mathbf{w};\boldsymbol{\lambda})P(\mathbf{w})} \tag{2.69}$$

where $p(\mathbf{O}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})$ is acoustic model likelihood (Section 2.3.2) and $P(\mathbf{w}_{1:L})$ is language model probability (Section 2.5). The language model in these discriminative criteria is not typically estimated in conjunction with the acoustic model so will be assumed fixed in this section.

2.7.1.1 Maximum mutual information

The HMM parameter estimation based on maximum mutual information (MMI) criterion [8, 164] can be performed by maximising [75]

$$\mathcal{F}_{\texttt{mmi}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log(P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})) \tag{2.70}$$

Re-writing equation (2.70) using the form of posterior in equation (2.69) yields

$$\mathcal{F}_{\texttt{mmi}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\frac{p(\mathbf{O}_{1:T_r}^{(r)}|\mathbf{w}_{1:L_r}^{(r)};\boldsymbol{\lambda})P(\mathbf{w}_{1:L_r}^{(r)})}{\sum_{\mathbf{w}} p(\mathbf{O}_{1:T_r}^{(r)}|\mathbf{w};\boldsymbol{\lambda})P(\mathbf{w})}\right) \tag{2.71}$$

where, leaving aside language model probabilities, *numerator* term is the likelihood of $\mathbf{O}_{1:T_r}^{(r)}$ given the reference transcription $\mathbf{w}_{1:L_r}^{(r)}$, whilst *denominator* term is the likelihood given all possible word sequences. Thus, the MMI objective function is maximised by making generation of $\mathbf{O}^{(r)}_{1:T_r}$ from the acoustic model associated with reference transcription $\mathbf{w}^{(r)}_{1:L_r}$ likely and from acoustic models associated with all other word sequences unlikely [75].

2.7.1.2 Minimum classification error

The HMM parameter estimation based on minimum classification error (MCE) criterion [35, 114] can be performed by minimising [75]

$$\mathcal{F}_{\texttt{mce}}(\boldsymbol{\lambda}, \xi; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left(1 + \left[\frac{P(\mathbf{w}^{(r)}_{1:L_r}|\mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\lambda})}{\sum\limits_{\mathbf{w}\neq\mathbf{w}^{(r)}_{1:L_r}} P(\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\lambda})}\right]^{\xi}\right)^{-1} \tag{2.72}$$

where ξ is an additional free parameter. Compared to the MMI objective function, the MCE objective function excludes the reference transcription from the denominator term and smooths the posteriors using a sigmoid-like function [75].

2.7.1.3 Minimum Bayes' risk

The HMM parameter estimation based on minimum Bayes' risk (MBR) criterion [26, 116] can be performed by minimising [75]

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\lambda})\mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{1:L_r}) \tag{2.73}$$

where $\mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{1:L_r})$ is a *loss* of word sequence $\mathbf{w}$ against the reference transcription $\mathbf{w}^{(r)}_{1:L_r}$. In contrast to MMI, which attempts to model the posterior distribution of reference transcriptions, the MBR criterion attempts to minimise the expected loss during decoding of the training data [75].

Designing a suitable loss function is crucial and leads to several variants of MBR criteria. For instance, the use of 0/1 or *sentence-level* loss function

$$\mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{1:L_r}) = \begin{cases} 0, & \text{if} \quad \mathbf{w} = \mathbf{w}^{(r)}_{1:L_r} \\ 1, & \text{if} \quad \mathbf{w} \neq \mathbf{w}^{(r)}_{1:L_r} \end{cases} \tag{2.74}$$

gives rise to the MCE objective function in equation (2.72), where free parameter ξ is fixed to 1 [75]. On the other hand, the use of *word-level* loss function yields *minimum word error* (MWE) criterion. The loss function would normally be computed by minimising the Levenshtein edit distance [75] - the same approach used to compute WER on decoding the test data (Section 2.6).

In practice it has been observed that reducing specificity in the loss function from word to phone level leads to a better generalisation on test data [184]. This variant is known as *minimum phone error* (MPE) criterion. However, the use of phone-level loss function reduces the number of possible errors to be corrected, compared to the number of observations, which may impact generalisation [75]. In order to address this issue, a *minimum phone frame error* (MPFE) criterion based on a smooth measure of the number of observations having incorrect phone label, known as Hamming distance [228], can be adopted [279].

2.7.1.4 Margin criteria

In addition to the MMI, MCE and MBR-type criteria, there has been recently interest in using margin-based criteria [107, 129, 137, 208]. According to statistical learning theory [244], a classifier with the largest margin, where margin is the smallest distance between examples of two classes, in general, yields the lowest generalisation error [244]. For training sequences, the distance between reference transcription (correct class) and an alternative transcription (incorrect class) can be expressed as the log-posterior ratio [180, 217]. Then, in the simplest case of margin maximisation (MM), the HMM parameter estimation can be performed by optimising [64]

$$\mathcal{F}_{\texttt{mm}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[\min_{\mathbf{w}\neq\mathbf{w}_{1:L_r}^{(r)}}\left\{\log\left(\frac{P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}{P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}\right)\right\}\right] \tag{2.75}$$

where margin is the term in squared brackets. The MM objective function has properties related to both MMI and MCE [64]: the log-posterior of reference transcription is included similar to MMI and denominator term does not include element representing reference transcription similar to MCE.

A range of margin-based criteria have been proposed extending upon the MM objective function [137, 185, 203, 208]. Rather than allowing the log-posterior ratio to grow arbitrary large, a minimum margin size constraint was introduced in [137]. The margin was required to be not smaller than a positive constant η. This can be accomplished by means of the following *hinge-loss* (HL) objective function to minimise [64]

$$\mathcal{F}_{\texttt{hl}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[\eta - \min_{\mathbf{w}\neq\mathbf{w}_{1:L_r}^{(r)}}\left\{\log\left(\frac{P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}{P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}\right)\right\}\right]_{+} \tag{2.76}$$

where $[\cdot]_+$ is the hinge-loss given by

$$[f(x)]_+ = \max(f(x), 0) \tag{2.77}$$

Alternatively, in [208] the margin was required to be not smaller than a Hamming (HA) distance. This led to the following form of HA objective function to minimise [64]

$$\mathcal{F}_{\texttt{ha}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[\max_{\mathbf{w}\neq\mathbf{w}_{1:L_r}^{(r)}}\left\{\mathcal{H}(\mathbf{w},\mathbf{w}_{1:L_r}^{(r)}) - \log\left(\frac{P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}{P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})}\right)\right\}\right]_{+} \tag{2.78}$$

where $\mathcal{H}(\mathbf{w},\mathbf{w}_{1:L_r}^{(r)})$ is the Hamming distance. Note that the same loss function was used in the MPFE criterion [279] discussed in Section 2.7.1.3. In order to simplify optimisation, the soft-max inequality [203]

$$\max_i x_i \leq \log\left(\sum_i \exp(x_i)\right) \tag{2.79}$$

was applied to yield the following upper bound [64]

$$\mathcal{F}_{\texttt{ha}}(\boldsymbol{\lambda};\mathcal{D}) \leq \frac{1}{R}\sum_{r=1}^{R}\Bigg[-\log\Big(P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})\Big)+ \\ \log\left(\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\lambda})\mathcal{L}(\mathbf{w},\mathbf{w}_{1:L_r}^{(r)})\right)\Bigg]_{+} \tag{2.80}$$

which was minimised by setting the loss function as follows [64]

$$\mathcal{L}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)}) = \begin{cases} \exp(\mathcal{H}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)})), & \text{if} \quad \mathbf{w} \neq \mathbf{w}_{1:L_r}^{(r)} \\ 0, & \text{if} \quad \mathbf{w} = \mathbf{w}_{1:L_r}^{(r)} \end{cases} \tag{2.81}$$

This upper bound has properties related to both the MMI and MBR criterion [64]. The first term within the hinge-loss function is the negated log-posterior, the same as the MMI objective function. The second term is the logarithm of MBR variant, where loss function is given by equation (2.81). Furthermore, this upper bound, if re-written in terms of acoustic model likelihood and language model probability, is related to a *boosted MMI* (bMMI) criterion [185]. The bMMI objective function to maximise can be expresses as [203]

$$\mathcal{F}_{\texttt{bmmi}}(\boldsymbol{\lambda}, \epsilon; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \log \left(\frac{p(\mathbf{O}_{1:T_r}^{(r)} | \mathbf{w}_{1:L_r}^{(r)}; \boldsymbol{\lambda}) P(\mathbf{w}_{1:L_r}^{(r)})}{\sum_{\mathbf{w}} p(\mathbf{O}_{1:T_r}^{(r)} | \mathbf{w}; \boldsymbol{\lambda}) P(\mathbf{w}) \exp(-\epsilon \mathcal{A}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)}))} \right) \tag{2.82}$$

where $\mathcal{A}(\mathbf{w}_{1:|\mathbf{w}|}, \mathbf{w}_{1:L_r}^{(r)})$ is phone-level accuracy. The difference between the upper bound and the bMMI objective function is the use of hinge-loss function in the former and the use of negated scaled phone-level accuracy function instead of the loss function in equation (2.81) in the latter.

2.7.1.5 Perceptron

The HMM parameter estimation based on the perceptron criterion [16, 49, 198] may be performed by minimising the following objective function [73]

$$\mathcal{F}_{\texttt{per}}(\boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \left[- \min_{\mathbf{w} \neq \mathbf{w}_{1:L_r}^{(r)}} \left\{ \log \left(\frac{P(\mathbf{w}_{1:L_r}^{(r)} | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\lambda})}{P(\mathbf{w} | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\lambda})} \right) \right\} \right]_+ \tag{2.83}$$

The perceptron criterion has properties related to margin criteria: it contains margin similar to MM and hinge loss function to prevent it from growing arbitrary large similar to HL. The perceptron criterion can be considered as a particular version of HL criterion where the margin is required to be not smaller than $\eta = 0$.

2.7.2 Optimisation of discriminative criteria

Compared to ML, the implementation of the discriminative criteria is more complicated [75]. For instance, although the MMI objective function in equation (2.71) contains two terms each resembling ML objective function, the Baum-Welch algorithm can not be used due to the minus sign between the two terms [184].

A number of optimisation schemes have been developed for discriminative criteria. For optimising MMI, MBR and MCE, these range from standard multivariate optimisation techniques [8, 154, 207] to an extension to the Baum-Welch algorithm (Section 2.2.3), known as *extended Baum-Welch algorithm* [7, 81, 84, 118, 172, 184, 207]. In addition, there are several options available for optimising margin-based criteria. For instance, the MM and H(L/A) objective functions can be optimised by means of a cutting plane algorithm [111, 238] to directly solve the constrained optimisation problems in equations (2.75), (2.76) and (2.78). The outcome of this approach is that the solution obtained will satisfy the minimum margin constraints for all training data pairs [277]. On the other hand, if a margin criterion can be reduced to a differentiable objective function, such as the lower bound to the HA objective function, then approaches such as sub-gradient method [211], exponentiated gradient method [78], [209] and, in case of the bMMI objective function, extended Baum-Welch algorithm [185, 203], become available.

The rest of this section will discuss the use of extended Baum-Welch algorithm for MMI and MPE estimation of HMM parameters. The presentation will adopt the concept of weak-sense auxiliary functions [184] to derive HMM parameter update rules, for an alternative perspective consider [7, 81, 84, 118, 172, 207].

2.7.2.1 Optimisation of MMI

As noted earlier, the Baum-Welch algorithm can not be applied to optimise MMI objective function - although an individual auxiliary function can be defined for the numerator, $\mathcal{Q}_{\texttt{num}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$, and the denominator term, $\mathcal{Q}_{\texttt{den}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$, their difference

$$\mathcal{G}_{\texttt{mmi}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \mathcal{Q}_{\texttt{num}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) - \mathcal{Q}_{\texttt{den}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) \tag{2.84}$$

is not a valid auxiliary function for the MMI objective function. Although $\mathcal{G}_{\mathtt{mmi}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ is not an auxiliary function in the *strong sense* [12, 45], it is a smooth function around the new set of parameters $\widehat{\boldsymbol{\lambda}}$ such that its gradient when evaluated at the current set of parameters $\boldsymbol{\lambda}$ equals to the gradient of the MMI objective function [184]

$$\nabla_{\widehat{\boldsymbol{\lambda}}}\, \mathcal{G}_{\mathtt{mmi}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})\Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} = \nabla_{\widehat{\boldsymbol{\lambda}}}\, \mathcal{F}_{\mathtt{mmi}}(\widehat{\boldsymbol{\lambda}}; \mathcal{D})\Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} \tag{2.85}$$

The function $\mathcal{G}_{\mathtt{mmi}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ is called a *weak-sense auxiliary function* for the MMI objective function $\mathcal{F}_{\mathtt{mmi}}(\widehat{\boldsymbol{\lambda}}; \mathcal{D})$ around the current set of parameters $\boldsymbol{\lambda}$. Maximising the weak-sense auxiliary function does not guarantee increase in the MMI objective function nor that it will converge, however, if it does converge then it will converge to the local maximum of the MMI objective function [184]. In order to improve convergence, the new set of parameters $\widehat{\boldsymbol{\lambda}}$ can be smoothed with the current set of parameters $\boldsymbol{\lambda}$ by means of a *smoothing function* $\mathcal{Q}_{\mathtt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}})$ as follows

$$\mathcal{G}_{\mathtt{mmi}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \mathcal{Q}_{\mathtt{num}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) - \mathcal{Q}_{\mathtt{den}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) + \mathcal{Q}_{\mathtt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}) \tag{2.86}$$

The smoothing function is required to have a zero gradient with respect to the new set of parameters $\widehat{\boldsymbol{\lambda}}$ when evaluated at the current set of parameters $\boldsymbol{\lambda}$ so not to affect the equality in equation (2.85). For optimising means and covariances, a suitable form is [55, 184]

$$\begin{aligned}\mathcal{Q}_{\mathtt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}) = -\frac{1}{2}\sum_{\{j,m\}} D_{j,m}\Big(&\log(|\widehat{\boldsymbol{\Sigma}}_{j,m}|) + \mathrm{tr}(\boldsymbol{\Sigma}_{j,m}\widehat{\boldsymbol{\Sigma}}_{j,m}^{-1}) + \\ &(\boldsymbol{\mu}_{j,m} - \widehat{\boldsymbol{\mu}}_{j,m})^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}_{j,m}^{-1}(\boldsymbol{\mu}_{j,m} - \widehat{\boldsymbol{\mu}}_{j,m})\Big)\end{aligned} \tag{2.87}$$

where summation is performed over all possible mixture components of all possible HMM states, $D_{j,m}$ is a state-component specific constant controlling the degree of smoothing. The value of this smoothing constant is crucial for optimisation and it is commonly set by

$$D_{j,m} = \max(2D_{j,m}^{\mathtt{low}}, E\gamma_{j,m}^{\mathtt{den}}) \tag{2.88}$$

where $D^{\texttt{low}}_{j,m}$ is the smallest value to ensure that $\widehat{\boldsymbol{\Sigma}}_{j,m}$ is positive-definite and E is a constant set in the range of 1 to 2 [184]. It can be shown that taking derivative of equation (2.86) and solving with respect to the new set of parameters $\widehat{\boldsymbol{\lambda}}$ yields the following update rules for the means and covariances [184]

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_{j,m} &= \frac{\left\{\boldsymbol{\theta}^{\texttt{num}}_{j,m} - \boldsymbol{\theta}^{\texttt{den}}_{j,m}\right\} + D_{j,m}\boldsymbol{\mu}_{j,m}}{\left\{\gamma^{\texttt{num}}_{j,m} - \gamma^{\texttt{den}}_{j,m}\right\} + D_{j,m}} && (2.89)\\ \widehat{\boldsymbol{\Sigma}}_{j,m} &= \frac{\left\{\boldsymbol{\Theta}^{\texttt{num}}_{j,m} - \boldsymbol{\Theta}^{\texttt{den}}_{j,m}\right\} + D_{j,m}(\boldsymbol{\Sigma}_{j,m} + \boldsymbol{\mu}_{j,m}\boldsymbol{\mu}^{\mathsf{T}}_{j,m})}{\left\{\gamma^{\texttt{num}}_{j,m} - \gamma^{\texttt{den}}_{j,m}\right\} + D_{j,m}} - \widehat{\boldsymbol{\mu}}_{j,m}\widehat{\boldsymbol{\mu}}^{\mathsf{T}}_{j,m} && (2.90)\end{aligned}$$

where the superscript `num` (and `den`) refers to quantities computed in composite HMMs constructed for the numerator (and denominator) term. These update rules are known as the extended Baum-Welch update rules [84, 89, 172, 184]. Compared to Baum-Welch update rules in equation (2.51) and (2.52), the extended Baum-Welch update rules in equation (2.89) and (2.90) also incorporate statistics derived from alternative word sequences.

As was mentioned at the beginning of Section 2.7, the use of MMI criterion may lead to over-training. In order to address this issue, a number of techniques have been developed to improve robustness of the estimates. One of them is an *acoustic de-weighting* [186], which is based on the use of the following form of posterior probability for accumulating the statistics [75]

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T};\boldsymbol{\lambda}) = \frac{p(\mathbf{O}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})^{\kappa}P(\mathbf{w}_{1:L})^{\beta}}{\sum\limits_{\mathbf{w}} p(\mathbf{O}_{1:T}|\mathbf{w};\boldsymbol{\lambda})^{\kappa}P(\mathbf{w})^{\beta}} \qquad (2.91)$$

Compared to the posterior in equation (2.69), the acoustic model likelihoods in acoustic de-weighting are raised to a fractional power κ, known as *acoustic de-weighting constant*. This makes less likely word sequences contribute more to the MMI objective function [184]. In practice, β is often set to one and κ is set to the inverse of the language model scale-factor [75].

Another technique to improve robustness of the estimates is an *I-smoothing technique* [187], which introduces a prior $p(\boldsymbol{\lambda};\boldsymbol{\lambda}^{\mathsf{p}})$ with hyper-parameters $\boldsymbol{\lambda}^{\mathsf{p}}$ on the HMM parameters into the MMI objective function. A modified form of the

MMI objective function can be expressed as [184]

$$\mathcal{F}_{\texttt{mmi}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(P(\mathbf{w}^{(r)}_{1:L_r}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\lambda})\right) + \log\left(p(\boldsymbol{\lambda};\boldsymbol{\lambda}^{\texttt{p}})\right) \tag{2.92}$$

which gives the following modified form of the weak-sense auxiliary function [184]

$$\mathcal{G}_{\texttt{mmi}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) = \mathcal{Q}_{\texttt{num}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) - \mathcal{Q}_{\texttt{den}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) + \mathcal{Q}_{\texttt{sm}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}}) + \log(p(\widehat{\boldsymbol{\lambda}};\widehat{\boldsymbol{\lambda}}^{\texttt{p}})) \tag{2.93}$$

The form of prior [55, 184] is the Normal-Wishart distribution [42]

$$\begin{aligned}\log(p(\widehat{\boldsymbol{\lambda}};\widehat{\boldsymbol{\lambda}}^{\texttt{p}})) = K - \frac{1}{2}\sum_{\{j,m\}}\tau^{\texttt{I}}\Big(&\log(|\widehat{\boldsymbol{\Sigma}}_{j,m}|)+ \\ &\mathrm{tr}((\widehat{\boldsymbol{\Sigma}}^{\texttt{p}}_{j,m}+\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m}\widehat{\boldsymbol{\mu}}^{\texttt{p}^{\mathsf{T}}}_{j,m})\widehat{\boldsymbol{\Sigma}}^{-1}_{j,m}) + (\widehat{\boldsymbol{\mu}}_{j,m}-\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m})^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}^{-1}_{j,m}(\widehat{\boldsymbol{\mu}}_{j,m}-\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m})\Big)\end{aligned} \tag{2.94}$$

where K is a constant, $\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m}$ and $\widehat{\boldsymbol{\Sigma}}^{\texttt{p}}_{j,m}$ are the prior mean and covariance, $\tau^{\texttt{I}}$ is an *I-smoothing constant* set around 100. One option to set the prior parameters is given by [184]

$$\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m} = \frac{\boldsymbol{\theta}^{\texttt{num}}_{j,m}}{\gamma^{\texttt{num}}_{j,m}} \tag{2.95}$$

$$\widehat{\boldsymbol{\Sigma}}^{\texttt{p}}_{j,m} = \frac{\boldsymbol{\Theta}^{\texttt{num}}_{j,m}}{\gamma^{\texttt{num}}_{j,m}} - \widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m}\widehat{\boldsymbol{\mu}}^{\texttt{p}^{\mathsf{T}}}_{j,m} \tag{2.96}$$

which yields ML-like estimates (see equation (2.51) and (2.52)) of mean and covariance that may change from iteration to iteration. In this case, the I-smoothing prior is known as *dynamic ML estimate* [265]. Taking derivative of equation (2.93) with respect to the new set of parameters $\widehat{\boldsymbol{\lambda}}$ yields the following update rules

$$\widehat{\boldsymbol{\mu}}_{j,m} = \frac{\left\{\boldsymbol{\theta}^{\texttt{num}}_{j,m}-\boldsymbol{\theta}^{\texttt{den}}_{j,m}\right\} + D_{j,m}\boldsymbol{\mu}_{j,m} + \tau^{\texttt{I}}\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m}}{\left\{\gamma^{\texttt{num}}_{j,m}-\gamma^{\texttt{den}}_{j,m}\right\} + D_{j,m} + \tau^{\texttt{I}}} \tag{2.97}$$

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_{j,m} &= \\ &\frac{\left\{\boldsymbol{\Theta}^{\texttt{num}}_{j,m}-\boldsymbol{\Theta}^{\texttt{den}}_{j,m}\right\}+D_{j,m}(\boldsymbol{\Sigma}_{j,m}+\boldsymbol{\mu}_{j,m}\boldsymbol{\mu}^{\mathsf{T}}_{j,m})+\tau^{\texttt{I}}(\widehat{\boldsymbol{\Sigma}}^{\texttt{p}}_{j,m}+\widehat{\boldsymbol{\mu}}^{\texttt{p}}_{j,m}\widehat{\boldsymbol{\mu}}^{\texttt{p}^{\mathsf{T}}}_{j,m})}{\left\{\gamma^{\texttt{num}}_{j,m}-\gamma^{\texttt{den}}_{j,m}\right\} + D_{j,m} + \tau^{\texttt{I}}} - \widehat{\boldsymbol{\mu}}_{j,m}\widehat{\boldsymbol{\mu}}^{\mathsf{T}}_{j,m}\end{aligned} \tag{2.98}$$

which apart from few additional terms are similar to the extended Baum-Welch update rules in equation (2.89) and (2.90).

Compared to accumulating statistics corresponding to the numerator term $\gamma_{j,m}^{\texttt{num}}$, $\boldsymbol{\theta}_{j,m}^{\texttt{num}}$ and $\boldsymbol{\Theta}_{j,m}^{\texttt{num}}$, known as *numerator statistics*, efficient handling of the denominator term is more complicated since this is equivalent to decoding the training data [75]. In order to address this issue, a *lattice-based framework* [184, 240] has been proposed, where statistics corresponding to the denominator term $\gamma_{j,m}^{\texttt{den}}$, $\boldsymbol{\theta}_{j,m}^{\texttt{den}}$ and $\boldsymbol{\Theta}_{j,m}^{\texttt{den}}$, known as *denominator statistics*, is accumulated by means of phone-marked lattices (Section 2.6.2). Given a phone-marked lattice $\mathbb{L}_{\texttt{den}}^{(r)}$, the denominator in equation (2.71) is approximated by lattice weight $[[\mathbb{L}_{\texttt{den}}^{(r)}]]$ defined earlier in equation (2.64). Given R phone-marked lattices, the denominator statistics can be expressed by [184]

$$\gamma_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \gamma_a \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \tag{2.99}$$

$$\boldsymbol{\theta}_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \gamma_a \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \mathbf{o}_t^{(r)} \tag{2.100}$$

$$\boldsymbol{\Theta}_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \gamma_a \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \mathbf{o}_t^{(r)} \mathbf{o}_t^{(r)\mathsf{T}} \tag{2.101}$$

where γ_a is the phone arc a occupancy and $\gamma_{a,j,m}(t)$ is the posterior probability to occupy phone arc a, state S_j and mixture component m at time t. The latter can be computed using the forward-backward algorithm (Section 2.2.2). The former can be computed using the lattice forward-backward algorithm (Section 2.6.2) where acoustic de-weighting is applied by raising HMM likelihoods to κ.

2.7.2.2 Optimisation of MPE

The MPE criterion was discussed in Section 2.7.1.3 as the variant of MBR criterion where loss is computed at the phone level. For implementation, it is common to use accuracy rather than the loss [184, 265]. The MPE objective function can be expressed as [187]

$$\mathcal{F}_{\texttt{mpe}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\lambda})\mathcal{A}(\mathbf{w},\mathbf{w}^{(r)}_{1:L_r}) \tag{2.102}$$

Using the form of posterior in equation (2.69) the MPE objective function can be written as

$$\mathcal{F}_{\texttt{mpe}}(\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\frac{\sum_{\mathbf{w}} p(\mathbf{O}^{(r)}_{1:T_r}|\mathbf{w};\boldsymbol{\lambda})P(\mathbf{w})\mathcal{A}(\mathbf{w},\mathbf{w}^{(r)}_{1:L_r})}{\sum_{\mathbf{w}} p(\mathbf{O}^{(r)}_{1:T_r}|\mathbf{w};\boldsymbol{\lambda})P(\mathbf{w})} \tag{2.103}$$

Compared to MMI objective function in equation (2.71), the MPE objective function in equation (2.103) is quite different. The form of weak-sense auxiliary function commonly used for optimising MPE objective function is also quite different from the weak-sense auxiliary function for MMI objective function in equation (2.84) [184]. The weak-sense auxiliary function for MPE objective function can be expressed in the lattice framework (Section 2.6.2) by [184]

$$\mathcal{Q}_{\texttt{mpe}}(\boldsymbol{\lambda},\hat{\boldsymbol{\lambda}};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \gamma^{\texttt{mpe}}_a \mathcal{Q}(\boldsymbol{\lambda},\hat{\boldsymbol{\lambda}};\mathcal{D}^{(r)}_{a,a^{\texttt{i}}}) \tag{2.104}$$

where $\mathcal{D}^{(r)}_{a,a^{\texttt{i}}} = \{\{\mathbf{O}^{(r)}_{\{a\}}, a^{\texttt{i}}\}\}$ is an observation sub-sequence and phone arc identity written in the form of supervised training data (see equation 2.42). The first term in equation (2.104) is the differential of MPE objective function with respect to HMM log-likelihood associated with phone arc

$$\gamma^{\texttt{mpe}}_a = \frac{\partial\mathcal{F}_{\texttt{mpe}}(\boldsymbol{\lambda};\mathcal{D})}{\partial\log(p(\mathbf{O}^{(r)}_{\{a\}}|a^{\texttt{i}};\boldsymbol{\lambda}))} \tag{2.105}$$

which can be both positive and negative. The second term in equation (2.104) is the (strong-sense) ML auxiliary function (Section 2.2.3). The differential can be expressed as [184]

$$\gamma^{\texttt{mpe}}_a = \gamma_a(c_a - c^{(r)}) \tag{2.106}$$

where c_a is the average accuracy of phone arc sequences passing phone arc a and $c^{(r)}$ is the average accuracy of all phone arc sequences in $\mathbb{L}^{(r)}_{\texttt{den}}$. These quantities can be efficiently computed by means of *forward* α'_a and *backward* β'_a *correctnesses* as follows [184]

$$\begin{aligned} c_a &= \alpha'_a + \beta'_a && (2.107) \\ c^{(r)} &= \alpha'_{a_{\mathcal{F}}} && (2.108) \end{aligned}$$

where $a_{\mathcal{F}}$ is the sentence end phone arc (Section 2.6.2). The forward and backward correctnesses can be computed recursively by [184]

$$\begin{aligned} \alpha'_a &= \frac{\sum\limits_{a' \text{ preceding } a} \alpha_{a'} P(a|a') \alpha'_{a'}}{\sum\limits_{a' \text{ preceding } a} \alpha_{a'} P(a|a')} + \widetilde{\mathcal{A}}(a) && (2.109) \\ \beta'_a &= \frac{\sum\limits_{a' \text{ following } a} P(a'|a) p(\mathbf{O}^{(r)}_{\{a'\}}|a'^{\mathtt{i}}; \boldsymbol{\lambda}) \beta_{a'} (\beta'_{a'} + \widetilde{\mathcal{A}}(a'))}{\sum\limits_{a' \text{ following } a} P(a'|a) p(\mathbf{O}^{(r)}_{\{a'\}}|a'^{\mathtt{i}}; \boldsymbol{\lambda}) \beta_{a'}} && (2.110) \end{aligned}$$

where α_a and β_a are the forward and the backward probability on the phone arc a (Section 2.6.2), $\widetilde{\mathcal{A}}(a)$ is the *approximate phone arc accuracy* given by [184]

$$\widetilde{\mathcal{A}}(a) = \max_{a' \in \mathbb{L}^{(r)}_{\texttt{num}}} \begin{cases} -1 + 2o(a, a'), & \text{if} \quad a^{\mathtt{i}} \equiv a^{\mathtt{i}'} \\ -1 + o(a, a'), & \text{otherwise} \end{cases} \qquad (2.111)$$

where $\mathbb{L}^{(r)}_{\texttt{num}}$ is the phone-marked lattice encoding the reference transcription $\mathbf{w}^{(r)}_{1:L_r}$, $o(a, a')$ is proportional to the overlap between $\mathbf{O}^{(r)}_{\{a\}}$ and $\mathbf{O}^{(r)}_{\{a'\}}$. In the case of total overlap, the approximate phone arc accuracy function returns 1 for a correct phone, 0 for a substitution and -1 for an insertion error [184]. In order to address possible over-training issues, the acoustic de-weighting can be applied by raising HMM likelihoods in equations (2.109) and (2.110) to κ (Section 2.7.2.1).

Given the weak-sense auxiliary function $\mathcal{Q}_{\texttt{mpe}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$, the smoothing function $\mathcal{Q}_{\texttt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}})$ in equation (2.87) and the prior $\log(p(\widehat{\boldsymbol{\lambda}}; \widehat{\boldsymbol{\lambda}}^{\mathtt{P}}))$ in equation (2.94) are added to form the final form of weak-sense auxiliary function

$$\mathcal{G}_{\texttt{mpe}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) = \mathcal{Q}_{\texttt{mpe}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D}) + \mathcal{Q}_{\texttt{sm}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}}) + \log(p(\widehat{\boldsymbol{\lambda}};\widehat{\boldsymbol{\lambda}}^{\texttt{p}})) \tag{2.112}$$

Given the final weak-sense auxiliary function $\mathcal{G}_{\texttt{mpe}}(\boldsymbol{\lambda},\widehat{\boldsymbol{\lambda}};\mathcal{D})$, the closed-form solutions in the form of extended Baum-Welch update equations (2.89) and (2.90) can be derived [184] by defining the numerator statistics by

$$\gamma^{\texttt{num}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t) \tag{2.113}$$

$$\boldsymbol{\theta}^{\texttt{num}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\mathbf{o}^{(r)}_{t} \tag{2.114}$$

$$\boldsymbol{\Theta}^{\texttt{num}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\mathbf{o}^{(r)}_{t}\mathbf{o}^{(r)^{\mathsf{T}}}_{t} \tag{2.115}$$

and the denominator statistics by

$$\gamma^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,-\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t) \tag{2.116}$$

$$\boldsymbol{\theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,-\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\mathbf{o}^{(r)}_{t} \tag{2.117}$$

$$\boldsymbol{\Theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0,-\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\mathbf{o}^{(r)}_{t}\mathbf{o}^{(r)^{\mathsf{T}}}_{t} \tag{2.118}$$

where phone arcs with positive differentials, $\gamma^{\texttt{den}}_{a} > 0$, provide numerator statistics and the remaining provide denominator statistics.

As noted in Section 2.7.2.1, in order to improve robustness of the estimates, in addition to acoustic de-weighting, it is possible to apply I-smoothing. For MPE training, the I-smoothing prior may be set to dynamic MMI estimate given by equations (2.97) and (2.98) which in turn makes use of dynamic ML estimate in equations (2.95) and (2.96) as its I-smoothing prior [265].

2.8 Adaptation to speaker and noise

In speech recognition it is common that test data includes poorly represented or completely new speakers and/or noise conditions [75]. For HMM-based approach such mismatch between training and test conditions is known to cause a degradation in recognition accuracy [74]. In order to address this mismatch, an *adaptation* is commonly performed where a small amount of data from a speaker and/or noise condition, called *adaptation data*, is used to modify the acoustic model or observations so as to more closely match that condition [75]. The adaptation can be applied both in *training* to reduce the level of variability present in the training data [65] and in *recognition* to reduce the mismatch and the consequent recognition errors [75]. Similar to the HMM parameter estimation problem (Section 2.7), it can be carried over in the *supervised* setting in which case reference are available, or in the *unsupervised* setting in which case they have to be hypothesised; in addition, adaptation is called *incremental* if the data comes in stages, or *batch-mode* if it is immediately available [75]. In this section only batch-mode adaptation in supervised or unsupervised setting is considered.

A range of adaptation approaches have been developed [65]: there are standard statistical techniques such as maximum-a-posteriori (MAP) [76]; some are based on general linear transforms [63, 133]; others are based on a model of how the mismatch impacts the acoustic models or observations [1, 2, 61, 131, 161]. *Maximum likelihood linear regression* transforms discussed in Section 2.8.1 are currently one of the most popular approaches to adapt to the speaker conditions [75, 265]. Being a general adaptation technique [65], they have also been applied to compensate the mismatch in noise conditions [74]. However, specific noise compensation approaches based on a model of how the noise impacts the acoustic models or observations are usually more effective, especially with very limited adaptation data [75]. When observation sequences are based on the MFCC feature extraction scheme (Section 2.1.1) then a *vector Taylor series* [2, 161] approach discussed in Section 2.8.2 can be used.

The rest of this section adopts bar notation to denote unmodified, *canonical*, acoustic models and unmodified, “clean”, observations. For instance, $\bar{\boldsymbol{\lambda}}$ denotes the canonical set of HMM parameters, whilst $\boldsymbol{\lambda}$ denotes the *adapted* set of HMM parameters. Similarly, $\bar{\mathbf{o}}$ denotes the "clean" observation, whilst $\mathbf{o}$ denotes the *noise-corrupted* observation.

2.8.1 Maximum likelihood linear regression

Various configurations of linear transforms have been proposed. In the simplest case, a global *maximum likelihood linear regression* (MLLR) transform may be applied to mean vectors [133]

$$\boldsymbol{\mu}_{j,m} = \mathbf{A}\overline{\boldsymbol{\mu}}_{j,m} + \mathbf{b} \tag{2.119}$$

where $\mathbf{A}$, $\mathbf{b}$ are transform parameters associated with mean vectors. This configuration is called *mean MLLR* [265]. In addition to mean vectors, it is also possible to adapt covariance matrices in which case [63]

$$\boldsymbol{\Sigma}_{j,m} = \mathbf{H}\overline{\boldsymbol{\Sigma}}_{j,m}\mathbf{H}^{\mathsf{T}} \tag{2.120}$$

where $\mathbf{H}$ are transform parameters associated with covariance matrices. This configuration is called *variance MLLR* [265]. When both mean vectors and covariance matrices are adapted then the state-component $q^{j,m}$ output density can be computed by transforming observations and mean vectors whilst keeping covariance matrices unchanged [75]

$$\begin{aligned} p(\mathbf{o}|q^{j,m}, \mathcal{T}) &= \mathcal{N}(\mathbf{o}; \mathbf{A}\overline{\boldsymbol{\mu}}_{j,m} + \mathbf{b}, \mathbf{H}\overline{\boldsymbol{\Sigma}}_{j,m}\mathbf{H}^{\mathsf{T}}) && (2.121)\\ &= |\mathbf{H}|^{-1}\mathcal{N}(\mathbf{H}^{-1}\mathbf{o}; \mathbf{H}^{-1}(\mathbf{A}\overline{\boldsymbol{\mu}}_{j,m} + \mathbf{b}), \overline{\boldsymbol{\Sigma}}_{j,m}) && (2.122)\end{aligned}$$

where $\mathcal{T}$ are transform parameters $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{H}$. Using this form it is possible to efficiently apply full transformations, especially in situations when covariance matrices are diagonal [265]. In addition, when the transformation matrices $\mathbf{A}$ and $\mathbf{H}$ are constrained to be the same then [63]

$$\begin{aligned} \boldsymbol{\mu}_{j,m} &= \mathbf{A}\overline{\boldsymbol{\mu}}_{j,m} + \mathbf{b} && (2.123)\\ \boldsymbol{\Sigma}_{j,m} &= \mathbf{A}\overline{\boldsymbol{\Sigma}}_{j,m}\mathbf{A}^{\mathsf{T}} && (2.124)\end{aligned}$$

then the state-component $q^{j,m}$ output density can be computed by transforming observations whilst keeping the means and covariances unmodified [75]

$$\begin{aligned} p(\mathbf{o}|q^{j,m}, \mathcal{T}) &= \mathcal{N}(\mathbf{o}; \mathbf{A}\overline{\boldsymbol{\mu}}_{j,m} + \mathbf{b}, \mathbf{A}\overline{\boldsymbol{\Sigma}}_{j,m}\mathbf{A}^{\mathsf{T}}) && (2.125)\\ &= |\mathbf{A}^{-1}|\mathcal{N}(\mathbf{A}^{-1}\mathbf{o} - \mathbf{A}^{-1}\mathbf{b}; \overline{\boldsymbol{\mu}}_{j,m}, \overline{\boldsymbol{\Sigma}}_{j,m}) && (2.126)\\ &= |\mathbf{A}^{-1}|\mathcal{N}(\overline{\mathbf{o}}; \overline{\boldsymbol{\mu}}_{j,m}, \overline{\boldsymbol{\Sigma}}_{j,m}) && (2.127) \end{aligned}$$

where $\overline{\mathbf{o}}$ is the transformed observation vector given by

$$\overline{\mathbf{o}} = \mathbf{A}^{-1}\mathbf{o} - \mathbf{A}^{-1}\mathbf{b} \tag{2.128}$$

This configuration is called *constrained MLLR* (CMLLR) [63]. Compared to mean and variance MLLR, the CMLLR does not require transforming means and covariances which makes this configuration efficient if the speaker (or environment) rapidly changes [75].

The following Section 2.8.1.1 discusses how the transform parameters $\mathcal{T}$ can be estimated. The use of adaptation in training for estimating the canonical sets of HMM parameters $\overline{\boldsymbol{\lambda}}$ is discussed in Section 2.8.1.2. Lastly, Section 2.8.1.3 discusses how multiple linear transforms can be incorporated rather than the global transform discussed so far.

2.8.1.1 Transform parameter estimation

All MLLR configurations discussed in this section require reference transcriptions of the adaptation data in order to obtain new transform parameters [75]. In contrast to supervised setting, in unsupervised setting these transcriptions are not given and must be inferred by the decoder [75]. In this case, adaptation is normally performed iteratively until convergence is achieved: given hypothesised transcriptions, the new transform parameters are estimated and used to decode the adaptation data to refine the hypothesised transcriptions [259]. This approach, however, requires computationally expensive decoding of the adaptation data to be performed several times and is sensitive to decoding errors since all words within hypotheses are treated equally probable [75]. An approach based on lattices can be adopted to obtain more robust to decoding errors transform parameters and avoid the need for re-decoding the adaptation data since these can be computationally efficiently re-scored [179].

Similar to the HMM parameter estimation problem discussed in Section 2.7, it is possible to estimate mean, variance and constrained MLLR transform parameters using maximum likelihood (ML) criterion [75]. The auxiliary function adopted with these configurations may be expressed as [265]

$$\mathcal{Q}(\mathcal{T}, \widehat{\mathcal{T}}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{\{j,m\}} \gamma_{j,m}^{(r)}(t) \log(\mathcal{N}(\mathbf{o}_t^{(r)}; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m})) + K \tag{2.129}$$

where $\mathcal{D}$ is the adaptation data, K is a constant subsuming terms not related to the state-component output densities. Compared to the ML auxiliary function in Section 2.2.3, there are some key differences. First, the state-component occupancies, $\gamma_{j,m}^{(r)}(t)$, are computed using the canonical set of HMM parameters $\overline{\boldsymbol{\lambda}}$ transformed by $\mathcal{T}$. Second, the new set of HMM parameters $\widehat{\boldsymbol{\lambda}}$ is obtained by transforming $\overline{\boldsymbol{\lambda}}$ with $\widehat{\mathcal{T}}$. Third, the auxiliary function is optimised with respect to the new transform parameters $\widehat{\mathcal{T}}$.

Whereas there are closed form solutions for mean MLLR, the variance and constrained MLLR configurations require an iterative solution [75] as comprehensively discussed in [63]. In addition to ML criterion, a range of discriminative criteria, such as MMI and MPE discussed in Section 2.7.1, have been examined with these MLLR configurations for supervised adaptation [84, 237, 239, 252, 253]. In unsupervised adaptation, the use of discriminative criteria was found to be more sensitive to errors in hypothesised transcriptions than in ML estimation and in practice is not commonly adopted [75, 84, 237, 253].

2.8.1.2 Speaker adaptive training

For *speaker independent* (SI) speech recognition, the training data necessarily includes a large number of speakers and hence results in the increased level of variability [75]. The use of adaptation in training, in the form of *speaker adaptive training* (SAT), has been proposed as one possible solution [4]. Figure 2.12 illustrates the concept behind SAT assuming that the training data contains Y speakers. For each speaker, an individual transform is estimated, given an initial

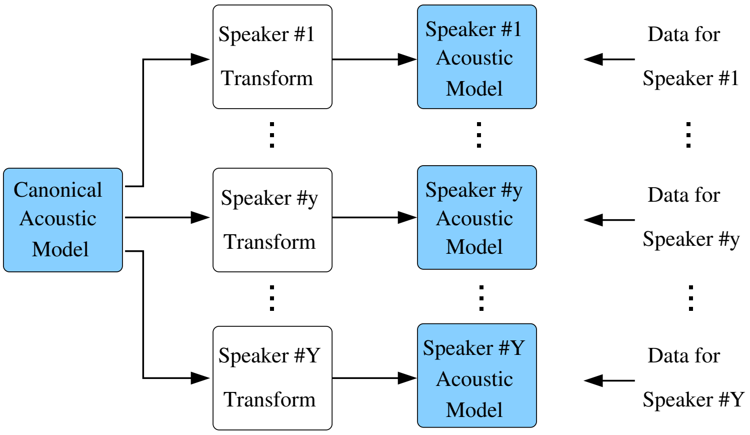


Figure 2.12: Speaker adaptive training.

estimate of canonical acoustic model and the training data available only to that speaker. The canonical acoustic model is re-estimated given Y transforms and the whole training data. This procedure is repeated until convergence is achieved or some maximum number of iterations reached.

Among MLLR configurations examined in this section, the use of CMLLR is most commonly adopted with SAT [75]. Similar to the HMM parameter estimation problem discussed in Section 2.7, it is possible to estimate canonical acoustic model parameters with CMLLR-based SAT using ML criterion [63]. In this case, the auxiliary function is given by

$$\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \sum_{y=1}^{Y} \sum_{r=1}^{R_y} \sum_{t=1}^{T_{y,r}} \gamma_{j,m}^{(y,r)}(t) \log(\mathcal{N}(\overline{\mathbf{o}}_t^{(y,r)}; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m})) + K \tag{2.130}$$

where R_y is the number of training observation sequences available to speaker y, $T_{y,r}$ is the number of observations in the r-th observation sequence available to speaker y, $\gamma_{j,m}^{(y,r)}(t)$ is the state-component occupancy computed using the current adapted set of HMM parameters (the current canonical set of HMM parameters $\overline{\boldsymbol{\lambda}}$ and speaker y transform parameters $\mathbf{A}^{(y)}$ and $\mathbf{b}^{(y)}$), the transformed observation vector $\overline{\mathbf{o}}_t^{(y,r)}$ defined by equation (2.128) is based on the training data observation vector $\mathbf{o}_t^{(y,r)}$ and speaker y transform parameters $\mathbf{A}^{(y)}$ and $\mathbf{b}^{(y)}$. The statistics required for updating canonical mean vectors $\overline{\boldsymbol{\mu}}_{j,m}$ and covariance matrices $\overline{\boldsymbol{\Sigma}}_{j,m}$

based on equation (2.51) and (2.52) respectively is given by [63]

$$\gamma_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{t=1}^{T_{y,r}} \gamma_{j,m}^{(y,r)}(t) \tag{2.131}$$

$$\boldsymbol{\theta}_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{t=1}^{T_{y,r}} \gamma_{j,m}^{(y,r)}(t)\overline{\mathbf{o}}_t^{(y,r)} \tag{2.132}$$

$$\boldsymbol{\Theta}_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{t=1}^{T_{y,r}} \gamma_{j,m}^{(y,r)}(t)\overline{\mathbf{o}}_t^{(y,r)}\overline{\mathbf{o}}_t^{(y,r)\mathsf{T}} \tag{2.133}$$

Apart from the use of transformed observation vectors, the statistics required are essentially the same as in Section 2.2.3.

In addition to ML criterion, a range of discriminative criteria, such as MMI and MPE discussed in Section 2.7.1, have been examined with CMLLR-based SAT [253]. The weak-sense auxiliary functions in both cases are essentially the same as in MMI (equation (2.93)) and MPE (equation (2.112)) estimation of HMM parameters yet the transformed observations in equation (2.128) are used, similar to the ML estimation of canonical model parameters above. The numerator and denominator statistics accumulated in both cases is essentially the same as in Section 2.7.2.1 yet the transformed observation vectors are used. For instance, the denominator statistics required for estimating the canonical mean vector $\overline{\boldsymbol{\mu}}_{j,m}$ and covariance matrix $\overline{\boldsymbol{\Sigma}}_{j,m}$ based on the extended Baum-Welch update rules in equations (2.97) and (2.98) in the MMI case is given by [253]

$$\gamma_{j,m}^{\mathtt{den}} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}_{\mathtt{den}}^{(y,r)}} \gamma_a \sum_{t\in\{a\}} \gamma_{a,j,m}(t) \tag{2.134}$$

$$\boldsymbol{\theta}_{j,m}^{\mathtt{den}} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}_{\mathtt{den}}^{(y,r)}} \gamma_a \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\overline{\mathbf{o}}_t^{(y,r)} \tag{2.135}$$

$$\boldsymbol{\Theta}_{j,m}^{\mathtt{den}} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}_{\mathtt{den}}^{(y,r)}} \gamma_a \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\overline{\mathbf{o}}_t^{(y,r)}\overline{\mathbf{o}}_t^{(y,r)\mathsf{T}} \tag{2.136}$$

and in the MPE case is given by [253]

$$\gamma^{\mathtt{den}}_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}^{(y,r)}_{\mathtt{den}}} \max(0,-\gamma^{\mathtt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t) \tag{2.137}$$

$$\boldsymbol{\theta}^{\mathtt{den}}_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}^{(y,r)}_{\mathtt{den}}} \max(0,-\gamma^{\mathtt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\overline{\mathbf{o}}^{(y,r)}_{t} \tag{2.138}$$

$$\boldsymbol{\Theta}^{\mathtt{den}}_{j,m} = \sum_{y=1}^{Y}\sum_{r=1}^{R_y}\sum_{a\in\mathbb{L}^{(y,r)}_{\mathtt{den}}} \max(0,-\gamma^{\mathtt{mpe}}_{a}) \sum_{t\in\{a\}} \gamma_{a,j,m}(t)\overline{\mathbf{o}}^{(y,r)}_{t}\overline{\mathbf{o}}^{(y,r)^{\mathsf{T}}}_{t} \tag{2.139}$$

where $\mathbb{L}^{(y,r)}_{\mathtt{den}}$ is the denominator lattice generated for the r-th observation sequence of speaker y. The numerator statistics for both cases is defined analogously.

In order to use these SAT canonical acoustic models for unsupervised adaptation on the test data, it is necessary to obtain initial hypothesised transcription or lattices as discussed in Section 2.8.1.1. Typically, a separate acoustic model trained on the whole training data is used to produce them [75]. In subsequent iterations of unsupervised adaptation they can be refined using estimated transform parameters and the SAT canonical acoustic models [55].

2.8.1.3 Regression classes

The use of global, per-speaker in the SAT case, transformation may not yield satisfactory gains in recognition accuracy. A powerful feature of linear transforms is that their number can be varied depending on the amount of adaptation data available [75]. For instance, when the amount of adaptation data is small then the use of global transform shared by all state-components may appear to be reasonable [75, 87]. As the amount of adaptation data increases, an individual transform may be associated with a *regression class* of state-components to give better adaptation [75]. When more data becomes available then some of the existing regression classes can be further split to give even better adaptation [133].

The number of regression classes to use with given adaptation data can be automatically determined using a *regression class tree* [62, 132], which is illustrated in Figure 2.13. Each node in the tree represents one regression class, i.e., a set of state-components that will share a single transform. For instance, the root node represents all state-components that will share a single global transform. Regression classes represented by the leaf nodes correspond to unique state-components and are known as *base classes* [62]. Thus, there are as many base classes (leaf nodes) as there are unique state-components.

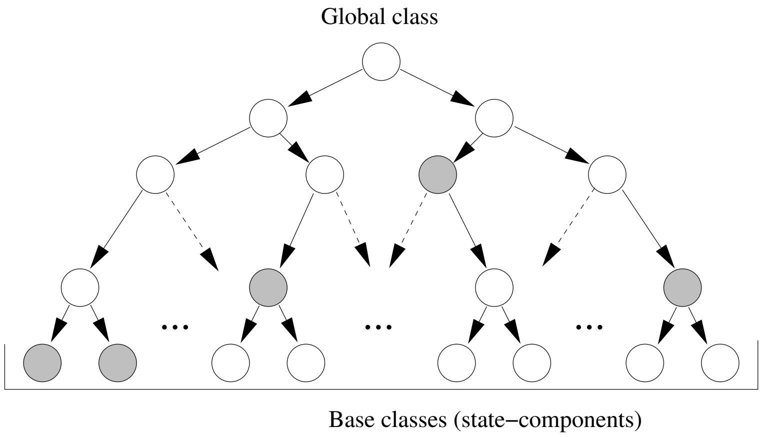


Figure 2.13: A regression class tree

Similar to phonetic decision trees discussed in Section 2.4, the total occupancy count associated with any node can be computed by accumulating the occupancy counts associated with its child leaf nodes. Once the total occupancy counts have been computed, the tree is descended to find the most specific set of nodes, such as those shown shaded in Figure 2.13, for which there is sufficient data and for which the transforms will be created.

Regression class trees may be built by making use of expert knowledge [87] or, more commonly, using automatic procedures which assume that state-components “close” to one another can share the same transform [132, 213, 265].

2.8.2 Vector Taylor series

The linear transforms examined in Section 2.8.1 are usually applied to address the mismatch in speaker conditions [75, 265]. Another very common and often extreme form of mismatch is cause by ambient noise [75]. Although these transforms can reduce the effects of noise [74], specific noise compensation schemes based on a *model* of how the noise impacts the acoustic models or observations can be more effective, especially with very limited adaptation data [75]. This section describes one such scheme called *vector Taylor series* (VTS).

There are many "noise" sources that may affect the "clean" speech signal such as stress, background noise, reverberation, the Lombard effect, microphone and transmission channel [61]. The VTS adopts a simplified model of the noisy acoustic environment [2] or *noise model*, which combines various additive and convolutional noise sources into single additive and linear channel or convolutional noises, as shown in Figure 2.14. This model assumes that the clean speech signal $\bar{o}$ is first subject to the convolutional noise h to which the additive noise n is added to yield the noise-corrupted speech signal o. The relationship between the clean and noise-corrupted o speech signals or the *mismatch function* may be written as [1]

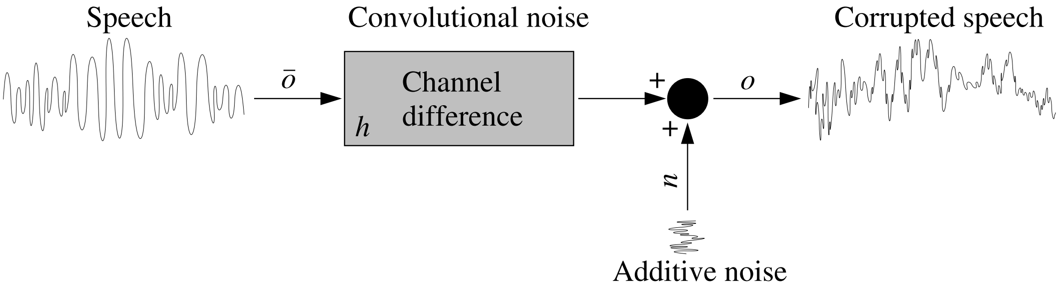


Figure 2.14: A simplified model of the noisy acoustic environment.

$$o = \bar{o} \otimes h + n \tag{2.140}$$

where $\otimes$ denotes convolution. As discussed in Section 2.1, speech signals are usually transformed into domains other than time. For instance, the mismatch function in equation (2.140) may be expressed in MFCC domain (Section 2.1.1) as [1, 2]

$$\mathbf{o}^{\mathsf{s}} = \mathbf{C}\log(\exp(\mathbf{C}^{-1}(\overline{\mathbf{o}}^{\mathsf{s}} + \mathbf{h}^{\mathsf{s}})) + \exp(\mathbf{C}^{-1}\mathbf{n}^{\mathsf{s}})) \tag{2.141}$$

where $\overline{\mathbf{o}}^{\mathsf{s}}$ and $\mathbf{o}^{\mathsf{s}}$ is the clean and noise-corrupted static[1] observation, $\mathbf{h}^{\mathsf{s}}$ is the con-

[1] The use of term *static* and superscript s refers everywhere in this thesis to the absence of dynamic information in the extracted observation features which typically (Section 2.1.2) come in the form of regression coefficients. When dynamic information is appended to the static observation vector $\mathbf{o}^{\mathsf{s}}$ then a complete observation vector $\mathbf{o}$ is formed.

volutional noise, $\mathbf{C}$ is the DCT matrix [140], $\mathbf{n}^{\mathsf{s}}$ is the additive noise, log and exp are element-wise logarithm and exponent functions. Given the mismatch function in equation (2.141), the noise-corrupted static mean, $\boldsymbol{\mu}^{\mathsf{s}}_{j,m}$, and covariance, $\boldsymbol{\Sigma}^{\mathsf{s}}_{j,m}$, for component $q^{j,m}$ may be expressed as [75]

$$\begin{aligned} \boldsymbol{\mu}^{\mathsf{s}}_{j,m} &= \mathcal{E}\{\mathbf{o}^{\mathsf{s}}|q^{j,m}\} & (2.142) \\ \boldsymbol{\Sigma}^{\mathsf{s}}_{j,m} &= \mathcal{E}\{\mathbf{o}^{\mathsf{s}}\mathbf{o}^{\mathsf{s}^{\mathsf{T}}}|q^{j,m}\} - \boldsymbol{\mu}^{\mathsf{s}}_{j,m}\boldsymbol{\mu}^{\mathsf{s}^{\mathsf{T}}}_{j,m} & (2.143) \end{aligned}$$

where $\mathcal{E}\{\cdot|q^{j,m}\}$ is the expectation taken with respect to the state-component $q^{j,m}$ output distribution. However, the mismatch function in equation (2.141) is highly *non-linear* which makes it complicated to derive closed-form expressions based on equations (2.142) and (2.143) [75].

The noise-corrupted observation $\mathbf{o}^{\mathsf{s}}$ based on equation (2.141) depends on the underlying clean observation $\overline{\mathbf{o}}^{\mathsf{s}}$, additive $\mathbf{n}^{\mathsf{s}}$ and convolutional $\mathbf{h}^{\mathsf{s}}$ noises. This can be expressed by means of function $\mathbf{f}$ which relates these variables

$$\mathbf{o}^{\mathsf{s}} = \mathbf{f}(\overline{\mathbf{o}}^{\mathsf{s}}, \mathbf{n}^{\mathsf{s}}, \mathbf{h}^{\mathsf{s}}) \tag{2.144}$$

In order to make the mismatch function $\mathbf{f}$ more manageable, the VTS applies *first-order vector Taylor series* expansion of $\mathbf{o}^{\mathsf{s}}$ for component $q^{j,m}$ around clean static mean $\overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m}$, additive noise mean $\boldsymbol{\mu}^{\mathsf{s,n}}$ and convolutional noise mean $\boldsymbol{\mu}^{\mathsf{s,h}}$ to linearise it [161]

$$\mathbf{o}^{\mathsf{s}}|q^{j,m} \approx \mathbf{f}(\overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m}, \boldsymbol{\mu}^{\mathsf{s,n}}, \boldsymbol{\mu}^{\mathsf{s,h}}) + (\overline{\mathbf{o}}^{\mathsf{s}} - \overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m})\frac{\partial \mathbf{f}}{\partial \overline{\mathbf{o}}^{\mathsf{s}}} + (\mathbf{n}^{\mathsf{s}} - \boldsymbol{\mu}^{\mathsf{s,n}})\frac{\partial \mathbf{f}}{\partial \mathbf{n}^{\mathsf{s}}} + (\mathbf{h}^{\mathsf{s}} - \boldsymbol{\mu}^{\mathsf{s,h}})\frac{\partial \mathbf{f}}{\partial \mathbf{h}^{\mathsf{s}}} \tag{2.145}$$

The partial derivatives or *Jacobians* above are evaluated at the expansion point and may be expressed by

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \overline{\mathbf{o}}^{\mathsf{s}}} &= \frac{\partial \mathbf{f}}{\partial \mathbf{h}^{\mathsf{s}}} = \mathbf{C}\mathbf{F}_{j,m}\mathbf{C}^{-1} = \mathbf{J}_{j,m} & (2.146) \\ \frac{\partial \mathbf{f}}{\partial \mathbf{n}^{\mathsf{s}}} &= \mathbf{I} - \mathbf{C}\mathbf{F}_{j,m}\mathbf{C}^{-1} = \mathbf{I} - \mathbf{J}_{j,m} & (2.147) \end{aligned}$$

where $\mathbf{F}_{j,m}$ is a diagonal matrix with elements given by $\mathbf{1} + \exp(\mathbf{C}^{-1}(\boldsymbol{\mu}^{\mathsf{s,n}} - \overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m} - \boldsymbol{\mu}^{\mathsf{s,h}}))$ [2]. Given the linearised form in equation (2.145), the noise-corrupted static parameters may be expressed as [2]

$$\begin{aligned}\boldsymbol{\mu}^{\mathsf{s}}_{j,m} &= \mathbf{f}(\overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m}, \boldsymbol{\mu}^{\mathsf{s,n}}, \boldsymbol{\mu}^{\mathsf{s,h}}) & (2.148)\\ \boldsymbol{\Sigma}^{\mathsf{s}}_{j,m} &= \mathbf{J}_{j,m}\overline{\boldsymbol{\Sigma}}^{\mathsf{s}}_{j,m}\mathbf{J}^{\mathsf{T}}_{j,m} + \mathbf{J}_{j,m}\boldsymbol{\Sigma}^{\mathsf{s,h}}\mathbf{J}^{\mathsf{T}}_{j,m} + (\mathbf{I}-\mathbf{J}_{j,m})\boldsymbol{\Sigma}^{\mathsf{s,n}}(\mathbf{I}-\mathbf{J}_{j,m})^{\mathsf{T}} & (2.149)\end{aligned}$$

where $\boldsymbol{\Sigma}^{\mathsf{s,n}}$ is the additive noise covariance matrix. The convolutional noise is commonly assumed to be constant, in which case $\boldsymbol{\Sigma}^{\mathsf{s,h}} = \mathbf{0}$ [2, 61, 161].

As discussed in Section 2.1.2, the complete observation vector $\mathbf{o}$ is formed by appending the dynamic coefficients $\Delta^{(1)}\mathbf{o}^{\mathsf{s}}, \ldots, \Delta^{(n)}\mathbf{o}^{\mathsf{s}}$ to the static observation vector $\mathbf{o}^{\mathsf{s}}$. In order to derive expressions for the dynamic parts, it is common to apply a *continuous time approximation* [82] under which the noise-corrupted first-order dynamic parameters can be expressed as [2]

$$\begin{aligned}\Delta^{(1)}\boldsymbol{\mu}^{\mathsf{s}}_{j,m} &= \mathbf{J}_{j,m}\Delta^{(1)}\overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m} & (2.150)\\ \Delta^{(1)}\boldsymbol{\Sigma}^{\mathsf{s}}_{j,m} &= \mathbf{J}_{j,m}\Delta^{(1)}\overline{\boldsymbol{\Sigma}}^{\mathsf{s}}_{j,m}\mathbf{J}^{\mathsf{T}}_{j,m} & (2.151)\end{aligned}$$

Similar expressions can be obtained for the higher-order dynamic parameters [2]. The complete noise-corrupted parameters are commonly set as follows [2, 140]

$$\boldsymbol{\mu}_{j,m} = \begin{bmatrix}\boldsymbol{\mu}^{\mathsf{s}\,\mathsf{T}}_{j,m} & \Delta^{(1)}\boldsymbol{\mu}^{\mathsf{s}\,\mathsf{T}}_{j,m} & \ldots & \Delta^{(n)}\boldsymbol{\mu}^{\mathsf{s}\,\mathsf{T}}_{j,m}\end{bmatrix}^{\mathsf{T}} \quad (2.152)$$

$$\boldsymbol{\Sigma}_{j,m} = \begin{bmatrix}\boldsymbol{\Sigma}^{\mathsf{s}}_{j,m} & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \Delta^{(1)}\boldsymbol{\Sigma}^{\mathsf{s}}_{j,m} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \ldots & \Delta^{(n)}\boldsymbol{\Sigma}^{\mathsf{s}}_{j,m}\end{bmatrix} \quad (2.153)$$

where the covariance terms are diagonalised for efficient decoding. Thus, the likelihood of $\mathbf{o}$ given state-component $q^{j,m}$ during decoding is given by [65]

$$p(\mathbf{o}|q^{j,m}) = \mathcal{N}(\mathbf{o}; \boldsymbol{\mu}_{j,m}, \text{diag}(\boldsymbol{\Sigma}_{j,m})) \quad (2.154)$$

The impact of this approximation will be discussed in the following.

The presentation so far has assumed that the noise $\boldsymbol{\lambda}^{\mathsf{n}} = \{\boldsymbol{\mu}^{\mathsf{n}}, \boldsymbol{\Sigma}^{\mathsf{n}}, \boldsymbol{\mu}^{\mathsf{h}}\}$ and clean acoustic model $\overline{\boldsymbol{\lambda}}$ parameters are given. In practice, these parameters are rarely known [65]. Although an estimate of the additive noise may be obtained from background, non-speech areas, this approach has several issues, such as the need for voice activity detector, sensitivity to changes in noise, inconsistency with the underlying acoustic model [140]. Furthermore, it is not straightforward to estimate the convolutional noise parameters from the background [65, 140]. In addition, it may not always be possible to directly estimate the clean acoustic model parameters [55]. A range of approaches have been proposed to estimate the noise model [66, 125, 136, 140, 161] and clean acoustic model [55, 66, 100, 117, 141] parameters. The following Section 2.8.2.1 adopts a *factor analysis* (FA) based approach [55, 66] which provides a consistent, EM-based, framework for estimating the noise and clean acoustic model parameters in Sections 2.8.2.2, 2.8.2.3 and 2.8.2.4.

2.8.2.1 Factor analysis generative models

Factor analysis (FA) is a statistical method for modelling the covariance structure of observed high-dimensional data using a small number of hidden variables or *factors* [55, 83, 200, 205]. For the noisy acoustic environment model in Figure 2.14, the observed data is the noise-corrupted static observation $\mathbf{o}^{\mathsf{s}}$ whilst the hidden variables are the clean static observation $\overline{\mathbf{o}}^{\mathsf{s}}$, the additive noise $\mathbf{n}^{\mathsf{s}}$ and the convolutional noise $\mathbf{h}^{\mathsf{s}}$. Given a state-component $q^{j,m}$, the FA generative model can be written as [66]

$$\mathbf{o}^{\mathsf{s}}|q^{j,m} = \boldsymbol{\Lambda}^{\mathsf{s,o}}_{j,m}\overline{\mathbf{o}}^{\mathsf{s}}|q^{j,m} + \boldsymbol{\Lambda}^{\mathsf{s,n}}_{j,m}\mathbf{n}^{\mathsf{s}} + \boldsymbol{\Lambda}^{\mathsf{s,h}}_{j,m}\mathbf{h}^{\mathsf{s}} + \boldsymbol{\epsilon}^{\mathsf{s}}_{j,m} \tag{2.155}$$

where

$$\overline{\mathbf{o}}^{\mathsf{s}}|q^{j,m} \sim \mathcal{N}(\overline{\boldsymbol{\mu}}^{\mathsf{s}}_{j,m}, \overline{\boldsymbol{\Sigma}}^{\mathsf{s}}_{j,m}) \tag{2.156}$$

$$\mathbf{n}^{\mathsf{s}} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathsf{s,n}}, \boldsymbol{\Sigma}^{\mathsf{s,n}}) \tag{2.157}$$

$$\mathbf{h}^{\mathsf{s}} \sim \delta(\mathbf{h}^{\mathsf{s}} - \boldsymbol{\mu}^{\mathsf{s,h}}) \tag{2.158}$$

$\boldsymbol{\Lambda}^{\mathtt{s,o}}_{j,m}$, $\boldsymbol{\Lambda}^{\mathtt{s,n}}_{j,m}$ and $\boldsymbol{\Lambda}^{\mathtt{s,h}}_{j,m}$ are *loading matrices*[1] associated with the clean static observation, additive and convolutional noise respectively, $\boldsymbol{\epsilon}^{\mathtt{s}}_{j,m} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathtt{s,e}}_{j,m}, \boldsymbol{\Sigma}^{\mathtt{s,e}}_{j,m})$ is a Gaussian distributed error term with mean and covariance given by

$$\boldsymbol{\mu}^{\mathtt{s,e}}_{j,m} = \boldsymbol{\mu}^{\mathtt{s}}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{s,o}}_{j,m}\overline{\boldsymbol{\mu}}^{\mathtt{s}}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{s,n}}_{j,m}\boldsymbol{\mu}^{\mathtt{s,n}} - \boldsymbol{\Lambda}^{\mathtt{s,h}}_{j,m}\boldsymbol{\mu}^{\mathtt{s,h}} \tag{2.159}$$

$$\boldsymbol{\Sigma}^{\mathtt{s,e}}_{j,m} = \boldsymbol{\Sigma}^{\mathtt{s}}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{s,o}}_{j,m}\overline{\boldsymbol{\Sigma}}^{\mathtt{s}}_{j,m}\boldsymbol{\Lambda}^{\mathtt{s,o}^{\mathsf{T}}}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{s,n}}_{j,m}\boldsymbol{\Sigma}^{\mathtt{s,n}}\boldsymbol{\Lambda}^{\mathtt{s,n}^{\mathsf{T}}}_{j,m} \tag{2.160}$$

A dynamic Bayesian network illustrating dependencies between observed and hidden variables in this generative model is shown in Figure 2.15. For this form of FA generative model, EM-based update formulae can be iteratively applied to obtain estimates of the additive noise and clean acoustic model parameters [83, 100, 122, 125, 200]. The convolutional noise mean can be estimated using a different EM-based approach [161, 201]. Together these update rules are guaranteed not to decrease the likelihood of generating the noise-corrupted data by the FA generative model [66].

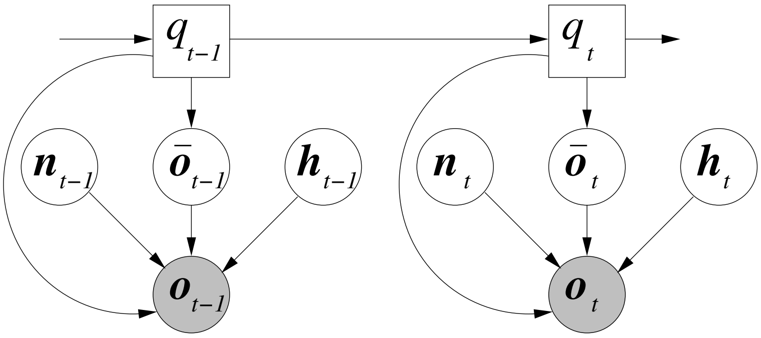


Figure 2.15: Dynamic Bayesian network for FA-style generative model.

The VTS mismatch function in equation (2.145) can be related to the FA generative model in equation (2.155) by re-writing it in the following way [66]

$$\mathbf{o}^{\mathtt{s}}|q^{j,m} \approx \mathbf{J}_{j,m}\overline{\mathbf{o}}^{\mathtt{s}}|q^{j,m} + (\mathbf{I} - \mathbf{J}_{j,m})\mathbf{n}^{\mathtt{s}} + \mathbf{J}_{j,m}\mathbf{h}^{\mathtt{s}} + \mathbf{g}^{\mathtt{s}}_{j,m} \tag{2.161}$$

[1] The rows of the loading matrices correspond to the observed variables and the columns correspond to the factors [39].

where

$$\mathbf{g}^{\mathtt{s}}_{j,m} = \boldsymbol{\mu}^{\mathtt{s}}_{j,m} - \mathbf{J}_{j,m}(\overline{\boldsymbol{\mu}}^{\mathtt{s}}_{j,m} + \boldsymbol{\mu}^{\mathtt{s},\mathtt{h}}) - (\mathbf{I} - \mathbf{J}_{j,m})\boldsymbol{\mu}^{\mathtt{s},\mathtt{n}} \tag{2.162}$$

The matrices $\mathbf{J}_{j,m}$ and $\mathbf{I} - \mathbf{J}_{j,m}$ are the partial derivatives or Jacobians of the VTS mismatch function (see equations 2.146 and 2.147). The Jacobians unlike the loading matrices $\boldsymbol{\Lambda}^{\mathtt{s},\mathtt{o}}_{j,m}$, $\boldsymbol{\Lambda}^{\mathtt{s},\mathtt{n}}_{j,m}$ and $\boldsymbol{\Lambda}^{\mathtt{s},\mathtt{h}}_{j,m}$ of the FA generative model depend on the clean static mean and noise model parameters (similar for the error terms). In addition, the covariance matrix of the noise-corrupted distribution based on equation (2.155) is generally full, unlike that used in the VTS case which is diagonalised for efficient decoding as discussed above. Both of these approximations mean that EM-based update formulae are not guaranteed to increase the likelihood of generating the noise-corrupted data [66]. In practice, the use of a back-off strategy helps to overcome these issues [140].

2.8.2.2 Noise estimation

The noise model parameters in this section are the convolutional noise mean $\boldsymbol{\mu}^{\mathtt{h}}$ and the additive noise mean $\boldsymbol{\mu}^{\mathtt{n}}$ and covariance $\boldsymbol{\Sigma}^{\mathtt{n}}$. As the convolutional noise is deterministic, it can be estimated in a standard fixed point approach [161]. In order to estimate the additive noise parameters, a simplified form of the FA generative model in equation (2.155) is used [66]

$$\mathbf{o}|q^{j,m} = \boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\mathbf{n} + \boldsymbol{\epsilon}_{j,m} \tag{2.163}$$

where the error term $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathtt{e}}_{j,m}, \boldsymbol{\Sigma}^{\mathtt{e}}_{j,m})$ parameters are fixed at the beginning to

$$\boldsymbol{\mu}^{\mathtt{e}}_{j,m} = \boldsymbol{\mu}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\boldsymbol{\mu}^{\mathtt{n}} \tag{2.164}$$

$$\boldsymbol{\Sigma}^{\mathtt{e}}_{j,m} = \boldsymbol{\Sigma}_{j,m} - \boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\boldsymbol{\Sigma}^{\mathtt{n}}\boldsymbol{\Lambda}^{\mathtt{n}^{\mathsf{T}}}_{j,m} \tag{2.165}$$

and the loading matrix $\boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}$ is forced to be diagonal. Note that compared to the previous section, the FA generative model is defined over complete observation vectors. The complete loading matrix in this case have the following form [261]

$$\boldsymbol{\Lambda}^{\mathtt{n}}_{j,m} = \mathrm{diag}\begin{bmatrix} \mathbf{I}-\mathbf{J}_{j,m} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-\mathbf{J}_{j,m} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}-\mathbf{J}_{j,m} \end{bmatrix} \tag{2.166}$$

where the number of blocks is equal to the order of regression.

Given observation sequence $\mathbf{O}_{1:T}$ and (hypothesised) reference transcription $\mathbf{w}_{1:L}$, the ML estimates of the additive noise model parameters can be found [65, 66]

$$\boldsymbol{\mu}^{\mathtt{n}} = \frac{\boldsymbol{\theta}^{\mathtt{n}}}{\gamma^{n}} \tag{2.167}$$

$$\boldsymbol{\Sigma}^{\mathtt{n}} = \mathrm{diag}\left(\frac{\boldsymbol{\Theta}^{\mathtt{n}}}{\gamma^{\mathtt{n}}} - \boldsymbol{\mu}^{\mathtt{n}}\boldsymbol{\mu}^{\mathtt{n}^{\mathsf{T}}}\right) \tag{2.168}$$

where the additive noise model statistics $\gamma^{\mathtt{n}}$, $\boldsymbol{\theta}^{\mathtt{n}}$ and $\boldsymbol{\Theta}^{\mathtt{n}}$ is given by

$$\gamma^{\mathtt{n}} = \sum_{t=1}^{T}\sum_{(j,m)} \gamma_{j,m}(t) \tag{2.169}$$

$$\boldsymbol{\theta}^{\mathtt{n}} = \sum_{t=1}^{T}\sum_{(j,m)} \gamma_{j,m}(t)\mathcal{E}\{\mathbf{n}_t|\mathbf{o}_t, q_t^{j,m}\} \tag{2.170}$$

$$\boldsymbol{\Theta}^{\mathtt{n}} = \sum_{t=1}^{T}\sum_{(j,m)} \gamma_{j,m}(t)\mathcal{E}\{\mathbf{n}_t\mathbf{n}_t^{\mathsf{T}}|\mathbf{o}_t, q_t^{j,m}\} \tag{2.171}$$

The expectations, which are taken over the noise-corrupted distribution determined by the current noise and clean acoustic model parameters, may be expressed as [66]

$$\mathcal{E}\{\mathbf{n}_t|\mathbf{o}_t, q_t^{j,m}\} = \boldsymbol{\mu}^{\mathtt{n}} + \boldsymbol{\Sigma}^{\mathtt{n}}\boldsymbol{\Lambda}^{\mathtt{n}^{\mathsf{T}}}_{j,m}\widetilde{\boldsymbol{\Sigma}}^{-1}_{j,m}(\mathbf{o}_t - \widetilde{\boldsymbol{\mu}}_{j,m}) = \boldsymbol{\mu}^{\mathtt{n|o}}_{j,m} \tag{2.172}$$

$$\mathcal{E}\{\mathbf{n}_t\mathbf{n}_t^{\mathsf{T}}|\mathbf{o}_t, q_t^{j,m}\} = \boldsymbol{\Sigma}^{\mathtt{n}} - \boldsymbol{\Sigma}^{\mathtt{n}}\boldsymbol{\Lambda}^{\mathtt{n}^{\mathsf{T}}}_{j,m}\widetilde{\boldsymbol{\Sigma}}^{-1}_{j,m}\boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\boldsymbol{\Sigma}^{\mathtt{n}} + \boldsymbol{\mu}^{\mathtt{n|o}}_{j,m}\boldsymbol{\mu}^{\mathtt{n|o}^{\mathsf{T}}}_{j,m} \tag{2.173}$$

where $\widetilde{\boldsymbol{\mu}}_{j,m}$ and $\widetilde{\boldsymbol{\Sigma}}_{j,m}$ are the noise-corrupted speech distribution parameters from the FA generative model

$$\tilde{\boldsymbol{\mu}}_{j,m} = \boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\boldsymbol{\mu}^{\mathtt{n}} + \boldsymbol{\mu}^{\mathtt{e}}_{j,m} \quad (2.174)$$

$$\tilde{\boldsymbol{\Sigma}}_{j,m} = \boldsymbol{\Lambda}^{\mathtt{n}}_{j,m}\boldsymbol{\Sigma}^{\mathtt{n}}\boldsymbol{\Lambda}^{\mathtt{n}^{\mathsf{T}}}_{j,m} + \boldsymbol{\Sigma}^{\mathtt{e}}_{j,m} \quad (2.175)$$

The additive noise mean may be initialised by setting $\boldsymbol{\mu}^{\mathtt{s,n}}$ equal to the static observation with the smallest energy [140, 161]. The additive noise covariance may be initialised by setting $\boldsymbol{\Sigma}^{\mathtt{s,n}}$ equal to the variance of the first (and possibly last) 3-10 static observations [124, 140].

2.8.2.3 VTS adaptive training

In order to estimate the clean acoustic model parameters, a simplified form of the FA generative model in equation (2.155) is used [66]

$$\mathbf{o}|q^{j,m} = \boldsymbol{\Lambda}^{\circ}_{j,m}\overline{\mathbf{o}} + \boldsymbol{\epsilon}_{j,m} \quad (2.176)$$

where the error term $\boldsymbol{\epsilon}_{j,m} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathtt{e}}_{j,m}, \boldsymbol{\Sigma}^{\mathtt{e}}_{j,m})$ parameters are fixed at the beginning to

$$\boldsymbol{\mu}^{\mathtt{e}}_{j,m} = \boldsymbol{\mu}_{j,m} - \boldsymbol{\Lambda}^{\circ}_{j,m}\overline{\boldsymbol{\mu}}_{j,m} \quad (2.177)$$

$$\boldsymbol{\Sigma}^{\mathtt{e}}_{j,m} = \boldsymbol{\Sigma}_{j,m} - \boldsymbol{\Lambda}^{\circ}_{j,m}\overline{\boldsymbol{\Sigma}}_{j,m}\boldsymbol{\Lambda}^{\circ^{\mathsf{T}}}_{j,m} \quad (2.178)$$

and the complete loading matrix $\boldsymbol{\Lambda}^{\circ}_{j,m}$ is given by [261]

$$\boldsymbol{\Lambda}^{\circ}_{j,m} = \mathrm{diag}\begin{bmatrix} \mathbf{J}_{j,m} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{j,m} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_{j,m} \end{bmatrix} \quad (2.179)$$

where the number of block is equal to the order of regression.

The training data $\mathcal{D}$ is assumed to be split into R noisy acoustic environment conditions, where each condition r is represented by one training observation sequence $\mathbf{O}^{(r)}_{1:T_r}$ [55, 65, 66]. The auxiliary function is given by [55, 123]

$$\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{\{j,m\}} \gamma_{j,m}(t) \mathcal{E}\{\mathcal{N}(\overline{\mathbf{o}}_t; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m}) | \mathbf{o}_t, q_t^{j,m}\} + K \tag{2.180}$$

where $\boldsymbol{\lambda}$ is the adapted current set of clean acoustic model parameters $\overline{\boldsymbol{\lambda}}$ and $\widehat{\boldsymbol{\lambda}}$ is the new set of clean acoustic model parameters. The ML estimates of the clean acoustic model parameters are given by equations (2.51) and (2.52), where the required statistics is [65, 66]

$$\begin{aligned}
\gamma_{j,m} &= \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma^{(r)}_{j,m}(t) && (2.181)\\
\boldsymbol{\theta}_{j,m} &= \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma^{(r)}_{j,m}(t) \mathcal{E}\{\overline{\mathbf{o}}^{(r)}_t | \mathbf{o}^{(r)}_t, q_t^{r,j,m}\} && (2.182)\\
\boldsymbol{\Theta}_{j,m} &= \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma^{(r)}_{j,m}(t) \mathcal{E}\{\overline{\mathbf{o}}^{(r)}_t \overline{\mathbf{o}}^{(r)^{\mathsf{T}}}_t | \mathbf{o}^{(r)}_t, q_t^{r,j,m}\} && (2.183)
\end{aligned}$$

The expectations, which are taken over the noise-corrupted distribution determined by the current noise and clean acoustic model parameters, may be expressed as [66]

$$\begin{aligned}
\mathcal{E}\{\overline{\mathbf{o}}^{(r)}_t | \mathbf{o}^{(r)}_t, q_t^{r,j,m}\} &= \overline{\boldsymbol{\mu}}_{j,m} + \overline{\boldsymbol{\Sigma}}_{j,m} \boldsymbol{\Lambda}^{\mathsf{o}^{\mathsf{T}}}_{j,m} \widetilde{\boldsymbol{\Sigma}}^{-1}_{j,m} (\mathbf{o}^{(r)}_t - \widetilde{\boldsymbol{\mu}}_{j,m}) = \boldsymbol{\mu}^{\mathsf{o}|\cdot}_{j,m} && (2.184)\\
\mathcal{E}\{\overline{\mathbf{o}}^{(r)}_t \overline{\mathbf{o}}^{(r)^{\mathsf{T}}}_t | \mathbf{o}^{(r)}_t, q_t^{r,j,m}\} &= \overline{\boldsymbol{\Sigma}}_{j,m} - \overline{\boldsymbol{\Sigma}}_{j,m} \boldsymbol{\Lambda}^{\mathsf{o}^{\mathsf{T}}}_{j,m} \widetilde{\boldsymbol{\Sigma}}^{-1}_{j,m} \boldsymbol{\Lambda}^{\mathsf{o}}_{j,m} \overline{\boldsymbol{\Sigma}}_{j,m} + \boldsymbol{\mu}^{\mathsf{o}|\cdot}_{j,m} \boldsymbol{\mu}^{\mathsf{o}|\cdot^{\mathsf{T}}}_{j,m} && (2.185)
\end{aligned}$$

where the noise-corrupted speech distribution parameters associated with the FA generative model are given by

$$\begin{aligned}
\widetilde{\boldsymbol{\mu}}_{j,m} &= \boldsymbol{\Lambda}^{\mathsf{o}}_{j,m} \overline{\boldsymbol{\mu}}_{j,m} + \boldsymbol{\mu}^{\mathsf{e}}_{j,m} && (2.186)\\
\widetilde{\boldsymbol{\Sigma}}_{j,m} &= \boldsymbol{\Lambda}^{\mathsf{o}}_{j,m} \overline{\boldsymbol{\Sigma}}_{j,m} \boldsymbol{\Lambda}^{\mathsf{o}^{\mathsf{T}}}_{j,m} + \boldsymbol{\Sigma}^{\mathsf{e}}_{j,m} && (2.187)
\end{aligned}$$

The initial estimates of the clean acoustic model parameters may be set to the ML estimates obtained as described in Section 2.2.3.

The noise and clean acoustic model parameters are usually estimated in a manner similar to speaker adaptive training discussed in Section 2.8.1.2, where given an initial estimate of the noise model parameters the clean acoustic model parameters are updated. Given new clean acoustic model parameters, the noise model parameters are updated and so on [55, 65, 66]. The clean acoustic model thus estimated is usually called *VTS adaptively trained* (VAT) acoustic model [55].

2.8.2.4 Discriminative VTS adaptive training

Another advantage of FA based approach is that it is possible to incorporate discriminative training criteria, such as minimum phone error (MPE) described in Section 2.7.1.3, into the VTS adaptive training framework [65]. For *discriminative VTS adaptive training* (DVAT), the noise model is assumed to be estimated as discussed in Section 2.8.2.4 and fixed. The MPE criterion in DVAT is optimised similar to Section 2.7.2.2 yet the statistics required by the EBW update rules in equation (2.97) and (2.98) in the denominator case is given by

$$\gamma_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \max(0, -\gamma_a^{\texttt{mpe}}) \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \tag{2.188}$$

$$\boldsymbol{\theta}_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \max(0, -\gamma_a^{\texttt{mpe}}) \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \mathcal{E}\{\overline{\mathbf{o}}_t^{(r)} | \mathbf{o}_t^{(r)}, q_t^{a,j,m}\} \tag{2.189}$$

$$\boldsymbol{\Theta}_{j,m}^{\texttt{den}} = \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\texttt{den}}^{(r)}} \max(0, -\gamma_a^{\texttt{mpe}}) \sum_{t \in \{a\}} \gamma_{a,j,m}(t) \mathcal{E}\{\overline{\mathbf{o}}_t^{(r)} \overline{\mathbf{o}}_t^{(r)\mathsf{T}} | \mathbf{o}_t^{(r)}, q_t^{a,j,m}\} \tag{2.190}$$

The numerator statistics has similar expressions. Other implementation details are comprehensively discussed in [55, 66].

2.9 Summary

This chapter gave an overview of hidden Markov model (HMM) based speech recognition. In particular, it described standard approaches to extract observations from speech, such as Mel-frequency cepstral coefficients. The use of HMMs for acoustic modelling and n-gram models for language modelling was discussed. In order to handle large number of possible sentence, the use of composite HMMs was discussed. In order to use these acoustic models for speech recognition, this section discussed how decoding and parameter estimation can be performed. In addition to maximum likelihood criterion, a range of discriminative criteria, such as maximum mutual information, minimum phone error and large margin, were discussed. Several approaches to ensure robustness of these estimates, such as phonetic decision tree based parameter tying and the use of smoothing during parameter estimation were discussed. In order to account for the mismatch between training and test conditions, the use of adaptation approaches, such as maximum likelihood linear regression and vector Taylor series, were discussed. The use of adaptive training, based on maximum likelihood and discriminative criteria, were detailed.

Chapter 3

Discriminative models

As was discussed in Chapter 1, the discriminative approach to speech recognition learns a direct map from observation sequences to sentences or models posterior probability of sentences given observation sequences directly [167]. This chapter discusses two discriminative models that have been applied to speech recognition tasks. The first discriminative model discussed in Section 3.1 is a *maximum entropy model* [14, 104, 199], which models the posterior probability of sentence labels given observation sequence. The second discriminative model discussed in Section 3.2 are *support vector machines* [68, 217, 244, 248], which in the simplest, binary case, map observation sequences into one of two sentence labels.

3.1 Maximum entropy models

Maximum entropy (MaxEnt) [103, 104] is a general technique for estimating probability distributions from training data [170]. Given a set of *knowledge sources* providing *statistics*, a combined model is created which imposes a set of *constraints* on the statistics to be satisfied [199]. The knowledge sources contribute to the statistics in the form of *feature-functions* $\boldsymbol{\phi}(\cdot)$ which extract fixed-dimensional *feature vectors* [14]. In case of speech recognition, these knowledge sources may be the acoustic and language model discussed in Chapter 2, the feature vectors are extracted from observation sequences $\mathbf{O}_{1:T}$ and sentence class labels ω which may be written as $\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)$ [73, 173]. Given a supervised training data consisting of

R training sequences

$$\mathcal{D} = \left\{ \{\mathbf{O}^{(1)}_{1:T_1}, \omega_1\}, \ldots, \{\mathbf{O}^{(R)}_{1:T_R}, \omega_R\} \right\} \tag{3.1}$$

the statistics may be expressed as the expected value of feature vectors with respect to the empirical distribution that generated the training data [170]

$$\mathcal{E}_{\mathcal{D}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)\} = \frac{1}{R}\sum_{r=1}^{R} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r}, \omega_r) \tag{3.2}$$

The usefulness of this statistics can be acknowledged by requiring the combined model to accord with it [14]. The expected value of feature vectors with respect to the combined model may be expressed as [170]

$$\mathcal{E}_{\boldsymbol{\alpha}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)\} \;=\; \frac{1}{R}\sum_{r=1}^{R}\sum_{\omega} P(\omega|\mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\alpha})\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r}, \omega) \tag{3.3}$$

where $\boldsymbol{\alpha}$ are the combined model parameters. The combined model may be set in accordance with the statistics by constraining [14, 170]

$$\mathcal{E}_{\boldsymbol{\alpha}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)\} = \mathcal{E}_{\mathcal{D}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)\} \tag{3.4}$$

Among all probability distributions $P(\omega|\mathbf{O}_{1:T}; \boldsymbol{\alpha})$ satisfying these constraints, the maximum entropy technique selects one which has the maximum entropy [103]. The unique solution to this problem maximises the posterior probability of reference sentence class labels in the training data and is a member of *exponential family* [170, 199]

$$P(\omega|\mathbf{O}_{1:T}; \boldsymbol{\alpha}) = \frac{1}{Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha})} \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega)) \tag{3.5}$$

where $Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha})$ is a *normalisation term* given by

$$Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha}) = \sum_{\omega'} \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega')) \tag{3.6}$$

The corresponding objective function to maximise may be expressed by [14, 170]

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(P(\omega_r|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha})\right) \tag{3.7}$$

This objective function is related to the maximum mutual information (MMI) objective function discussed in Section 2.7.1.1 and in the context of discriminative models is usually called *conditional maximum likelihood* (CML) [73]. The combined model with posterior probabilities defined as outlined above is called *maximum entropy model* [14]. Furthermore, discriminative models adopting the form of posterior probability in equation (3.5) are also known as *log-linear models* [90, 170, 199] and *flat direct models* [168].

Given observation sequence $\mathbf{O}_{1:T}$ and parameters $\boldsymbol{\alpha}$, inferring the most likely sentence class label with MaxEnt uses Bayes' decision rule [73]

$$\begin{aligned}\widehat{\omega} &= \arg\max_{\omega}\{P(\omega|\mathbf{O}_{1:T};\boldsymbol{\alpha})\} & (3.8)\\ &= \arg\max_{\omega}\left\{\frac{1}{Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\omega))\right\} & (3.9)\end{aligned}$$

Since $Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})$ is constant for each ω in equation (3.9) and does not change rank ordering then inferring can be equivalently performed by searching for ω with the largest dot-product of parameters $\boldsymbol{\alpha}$ and feature vector $\boldsymbol{\phi}(\mathbf{O}_{1:T},\omega)$ [73]

$$\widehat{\omega} = \arg\max_{\omega}\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\omega)\} \tag{3.10}$$

If posterior $P(\omega|\mathbf{O}_{1:T};\boldsymbol{\alpha})$ is required, such as for system combination purposes, it can be obtained using equation (3.5) once dot-products in equation (3.10) have been computed for all possible ω [128].

The rest of this section is organised as follows. The next Section 3.1.1 discusses feature-functions. The following Section 3.1.2 then discusses options available for estimating parameters. Adaptation to speaker and noise conditions is discussed last in Section 3.1.3.

3.1.1 Feature-functions

The form of feature-functions is central to the performance of MaxEnt [73]. For speech recognition, a wide range of feature-functions have been proposed as will be exemplified in Chapter 6. A fundamental requirement imposed on all those feature-functions is that they transform variable length sequences into a *fixed-length* representation (feature vectors) as multiple different length sequences may be used by the same feature-functions [73]. An example of feature-function extracting features only from observation sequences is given by

$$\boldsymbol{\phi}(\mathbf{O}_{1:T}) = \left[\sum_{t=1}^{T} \mathbf{o}_t \right] \tag{3.11}$$

Though simple, it bears a resemblance to the HMM mean statistics in equation (2.49) (see Section 2.2.3). As these features do not depend on sentence class labels it is necessary to explicitly relate observation sequences with sentence class label. The following mapping can be used to accomplish this

$$\boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega) = \begin{bmatrix} \vdots \\ \delta(\omega, \texttt{the dog chased the cat})\boldsymbol{\phi}(\mathbf{O}_{1:T}) \\ \vdots \end{bmatrix} \tag{3.12}$$

With d-dimensional observation vectors and $|\Omega|$ sentence class labels the total dimensionality of feature vectors in equation (3.12) is $d|\Omega|$. Thus, the number of parameters in such model would also be $d|\Omega|$.

In addition to mapping variable length sequences to fixed length, these feature-functions impose the first-order relationship between observations and sentence class labels [73]. For discriminative models it is common to represent such relationship by means of graphical models [112]. Figure 3.1 shows an example of graphical model associated with these feature-functions. The relationship or *dependencies* between individual observation vectors and the sentence class label have no direction and hence it is called *undirected graphical model* [112].

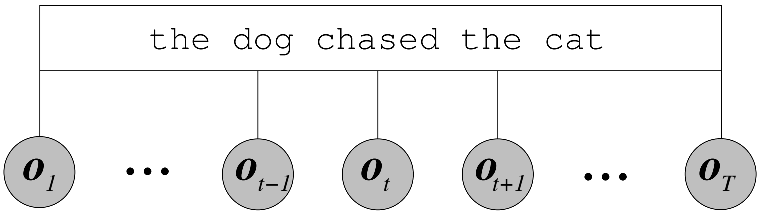


Figure 3.1: An example of graphical model for a simple discriminative model where feature functions establish the first-order relationship between observation sequences and sentence class labels.

3.1.2 Parameter estimation

As discussed at the beginning of this section, the MaxEnt parameters are estimated by optimising the CML objective function in equation (3.7), which is concave, having a single global maximum [170]. For optimisation, it is possible to use standard multi-dimension optimisation techniques [171, 195], such as gradient ascent, conjugate gradient and Rprop [169, 170], as well as specific algorithms such as generalised iterative scaling [40, 44] and improved iterative scaling [14, 44, 170]. The former approach have been found [149, 155, 280] to have faster convergence in many cases and hence will be adopted in this section. In addition to CML, it is also possible to use any of discriminative criteria discussed with HMMs in in Section 2.7.1 [73]. The rest of this section as an example discusses optimisation of CML, minimum Bayes' risk (MBR) and margin-based objective functions in Section 3.1.2.1, 3.1.2.2 and 3.1.2.3 respectively.

3.1.2.1 Optimisation of CML

The CML objective function to be optimised was given earlier in equation (3.7). The gradient of it with respect to the discriminative model parameters $\boldsymbol{\alpha}$ may be expressed in terms of the expected values in equation (3.2) and (3.3) as [169]

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha};\mathcal{D}) = \mathcal{E}_{\mathcal{D}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T},\omega)\} - \mathcal{E}_{\boldsymbol{\alpha}}\{\boldsymbol{\phi}(\mathbf{O}_{1:T},\omega)\} \tag{3.13}$$

Having computed gradient, the initial discriminative model parameters $\boldsymbol{\alpha}$ may be re-estimated using gradient-based optimisation techniques such as those discussed above. For instance, the use of gradient ascent yields the following update rule

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \eta \nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) \tag{3.14}$$

where η is a step size which in the simplest case is set to a small constant value.

Similar to discriminative estimation of HMM parameters, the CML estimation of maximum entropy model parameters has a tendency to over-train on the training data, i.e., it does not generalise well to the test data [80]. In order to address this issue, a *Gaussian prior* on the discriminative model parameters may be introduced [33]

$$P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\mathtt{p}}) = \mathcal{N}(\boldsymbol{\alpha}; \boldsymbol{\mu}^{\mathtt{p}}, \boldsymbol{\Sigma}^{\mathtt{p}}) \tag{3.15}$$

where $\boldsymbol{\alpha}^{\mathtt{p}} = (\boldsymbol{\mu}^{\mathtt{p}}, \boldsymbol{\Sigma}^{\mathtt{p}})$ are the prior parameters. Usually, the mean vector $\boldsymbol{\mu}^{\mathtt{p}}$ is set to zero and the covariance matrix $\boldsymbol{\Sigma}^{\mathtt{p}}$ has a diagonal form [168]

$$\boldsymbol{\Sigma}^{\mathtt{p}} = \sigma^{\mathtt{p}} \mathbf{I} \tag{3.16}$$

which introduces only one variance parameter or [170]

$$\boldsymbol{\Sigma}^{\mathtt{p}} = \begin{bmatrix} \sigma^{\mathtt{p}}_{1,1} & 0 & \dots & 0 \\ 0 & \sigma^{\mathtt{p}}_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^{\mathtt{p}}_{\dim(\boldsymbol{\alpha}),\dim(\boldsymbol{\alpha})} \end{bmatrix} \tag{3.17}$$

which introduces an individual variance parameter for each discriminative model parameter. The final objective function may be expressed as [80, 170]

$$\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D}) = \mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) + \log(P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\mathtt{p}})) \tag{3.18}$$

The Gaussian prior (and its logarithm) is concave which makes the final objective function to be concave and have a single global maximum [170]. The gradient of $\log(\mathcal{N}(\boldsymbol{\alpha}; \boldsymbol{\mu}^{\mathtt{p}}, \boldsymbol{\Sigma}^{\mathtt{p}}))$ with respect to discriminative model parameters is given by

$$\nabla_{\boldsymbol{\alpha}} \log(\mathcal{N}(\boldsymbol{\alpha}; \boldsymbol{\mu}^{\mathtt{p}}, \boldsymbol{\Sigma}^{\mathtt{p}})) = -\boldsymbol{\Sigma}^{\mathtt{p}^{-1}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\mathtt{p}}) \tag{3.19}$$

The use of Gaussian prior leads to the following form of update rule in, for instance, gradient ascent optimisation technique

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \eta \left(\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) - \boldsymbol{\Sigma}^{\mathrm{P}^{-1}} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\mathrm{P}}) \right) \tag{3.20}$$

In addition or as an alternative to Gaussian priors, it is possible use exponential and Laplacian priors [30, 80].

3.1.2.2 Optimisation of MBR

As in the previous section, the MBR objective function to be minimised may be obtained from the related objective function in equation (2.73) discussed in the context of MBR estimation of HMM parameters as

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\omega} P(\omega | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}) \mathcal{L}(\omega, \omega_r) \tag{3.21}$$

where $\mathcal{L}(\omega, \omega_r)$ is the loss between sentence class label ω and the r-th reference sentence class label ω_r. Replacing the loss $\mathcal{L}(\omega, \omega_r)$ with accuracy function $\mathcal{A}(\omega, \omega_r)$ and changing the direction of optimisation yields a similar to equation (2.102) form of MBR objective function

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\omega} P(\omega | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}) \mathcal{A}(\omega, \omega_r) \tag{3.22}$$

Following [128], the gradient with respect to parameters $\boldsymbol{\alpha}$ may be expressed as

$$\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) = \tag{3.23}$$
$$\frac{1}{R} \sum_{r=1}^{R} \sum_{\omega} P(\omega | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}) \left(\mathcal{A}(\omega, \omega_r) - \sum_{\omega'} P(\omega' | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}) \mathcal{A}(\omega', \omega_r) \right) \boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega)$$

This expression is related to the expected value in equation (3.3) which defines the gradient of CML objective function in equation (3.13). In addition, the posterior probability weighted term in parentheses above is related to the MPE differential in equation (2.106). Having computed the gradient, the initial parameters $\boldsymbol{\alpha}$ may be re-estimated using, for example, the gradient ascent optimisation technique discussed in the previous section.

In contrast to CML objective function, the MBR objective function in equation (3.21) or (3.22), is not typically concave, which makes it sensitive to initialisation and spurious local maxima [47]. Following [275], the use of CML estimate may be suggested to provide better initial parameters. In order to avoid local maxima, annealing techniques may be adopted [215]. In addition, similar to CML estimation, the MBR estimation has a tendency to over-train on the training data, i.e., it does not generalise well to the test data [215]. In order to address this issue, it is similarly possible to introduce Gaussian prior [47, 215].

3.1.2.3 Optimisation of large margin

A range of objective functions were discussed in the context of margin-based estimation of HMM parameters in Section 2.7.1.4. For maximum entropy models, one suitable form of objective function to minimise may be expressed as [278]

$$\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[\max_{\omega\neq\omega_r}\left\{\mathcal{L}(\omega,\omega_r) - \log\left(\frac{P(\omega_r|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha})}{P(\omega|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha})}\right)\right\}\right]_{+} \tag{3.24}$$

Depending on the form of loss function used it may related to the objective function in equation (2.76) (constant loss) or equation (2.76) (Hamming loss) discussed in the context of margin-based estimation of HMM parameters.

Using the form of posterior probability in equation (3.5) the large margin objective function may be expressed as [278]

$$\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[-\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\omega_r) + \max_{\omega\neq\omega_r}\left\{\mathcal{L}(\omega,\omega_r) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\omega)\right\}\right]_{+} \tag{3.25}$$

Note that the normalisation term $Z(\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha})$ common to the numerator and denominator term in equation (3.24) cancels out [278]. The objective function in equation (3.25) is convex (maximum of affine functions is a convex function [228]), having a single global minimum [278].

In order to address possible over-training issues, a prior may be introduced

[228]. For instance, the use of Gaussian prior given by equation (3.15) with zero mean vector $\boldsymbol{\mu}^{\mathtt{p}} = \mathbf{0}$ and scaled diagonal covariance matrix $\boldsymbol{\Sigma}^{\mathtt{p}} = \sigma^{\mathtt{p}}\mathbf{I}$ yields, after omitting terms constant in $\boldsymbol{\alpha}$, the following final form of the large margin objective function [278]

$$\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 + \sigma^{\mathtt{p}}\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}; \mathcal{D}) \tag{3.26}$$

where $\| \cdot \|_2$ is the Euclidean norm, also known as the ℓ_2 norm [49]

$$\|\boldsymbol{\alpha}\|_2 = \sqrt{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\alpha}} \tag{3.27}$$

The final objective function is convex (squared Euclidean distance is convex) and may be minimised using approaches discussed in [278]. The same approaches may be adopted when Gaussian prior has non-zero mean vector [275].

3.1.3 Adaptation to speaker and noise

In order to work reliably in real-world applications any speech recognition system must be designed to be robust to changes in speaker and noise conditions [75]. For HMMs, an overview of adaptation to the speaker and noise conditions was given in Section 2.8. A range of related approaches have also been developed for maximum entropy models [28, 147, 221, 222].

The use of maximum-a-posteriori (MAP) adaptation has been considered in [28]. It yields a general adaptation scheme which makes no assumption about the nature of the feature-functions [73]. However, when the feature-functions exhibit some structure the use of other adaptation schemes may be more advantageous [73].

Alternatively, the use of linear transformation based adaptation has been investigated in [147, 227]. These schemes make use of approaches similar to the maximum likelihood linear regression (MLLR) for HMMs discussed in Section 2.8.1. In contrast to the MAP adaptation, these schemes make assumptions about the relationships between features [73]. As these schemes have been applied only with feature-functions resembling those used in standard HMMs and also those given by equation (3.12), it is not clear whether this form of adaptation approaches can be extended to more general feature-functions [73].

Finally, the feature-functions can be modified to make them dependent on the speaker and noise conditions [73]. Related approaches have also been for HMMs [18, 46, 58, 131, 162, 202, 249]. The maximum entropy models then can be trained speaker and noise independent using those modified feature-functions [68]. This approach will be discussed in Chapter 6.

3.2 Support vector machines

The support vector machines (SVM) [243] is a discriminative classifier approximately implementing structural risk minimisation and based upon the intuitive concept of margin maximisation - first discussed in the context of discriminative estimation of hidden Markov model (HMM) parameters in Section 2.7.1.4 [72, 128]. In its standard implementation [109, 243, 244], the SVMs are a binary classifier of fixed-length vectors that have been applied to a range of tasks such as text categorisation [108], image recognition [37, 178], bioinformatics [86] and medical applications [50]. This standard SVM implementation is outlined next in Section 3.2.1. Section 3.2.2 then discusses an approach how variable rather than fixed-length vectors, such as observation sequences, may be handled. The following Section 3.2.3 discusses how situations with more than two classes - common to speech recognition [72] - may be addressed. Finally, the aspect of adapting classifier to mismatches between training and test data conditions - common to speech recognition [72] - is discussed in Section 3.2.4.

3.2.1 Standard implementation

Consider a training set

$$\mathcal{D} = \{\{\mathbf{o}_1, y_1\}, \dots, \{\mathbf{o}_R, y_R\}\} \tag{3.28}$$

where $\mathbf{o}_r$ is a d-dimensional vector and y_r is a binary, positive ($y_r = 1$) or negative ($y_r = -1$), class label for each r. The training set $\mathcal{D}$ is said to be *linearly separable* if there exists a vector $\mathbf{m}$ (weight vector) and a scalar b (bias) such that

$$y_r(\mathbf{m} \cdot \mathbf{o}_r + b) \geq 1 \tag{3.29}$$

is valid for each r [37, 243, 244]. The vectors for which equality in equation (3.29) holds are called *support vectors* [37]. Given a linearly separable training set, a hyperplane

$$\mathbf{m} \cdot \mathbf{o} + b = 0 \tag{3.30}$$

is said to be *optimal* if it separates the training set with a maximal margin between the vectors of the two classes [37], where the *margin* is defined as the shortest distance from a hyperplane to the closest positive and negative class vector [25]

$$\text{margin} = \frac{|1-b|}{\|\mathbf{m}\|_2} + \frac{|-1-b|}{\|\mathbf{m}\|_2} = \frac{2}{\|\mathbf{m}\|_2} \tag{3.31}$$

As was mentioned in Section 2.7.1.4, a classifier with large margin can be expected to yield good generalisation. In order to maximise the margin, the following constrained optimisation problem may be formulated [244]

$$\widehat{\mathbf{m}} = \arg\min_{\mathbf{m}} \left\{ \frac{1}{2} \|\mathbf{m}\|_2^2 \right\} \tag{3.32}$$

subject to the constraints in equation (3.29). Classification of test vectors can be performed based on the following form of decision function [37]

$$\widehat{y} = \text{sgn}(\mathbf{m} \cdot \mathbf{o} + b) \tag{3.33}$$

Figure 3.2 shows a linearly separable training set, where inequality in equation (3.29) is satisfied everywhere on the right/left of the right/left most dashed hyperplane for positive/negative class vectors, the support vectors are enclosed in squares, the solid hyperplane is the optimal hyperplane and the two-headed arrow is the maximal margin.

For a linearly non-separable training set, the optimisation problem in equations (3.32) has no feasible solution [25]. In order to address this issue, non-negative *slack variables* $\xi_1, \ldots, \xi_R$ may be introduced to formalise margin viola-

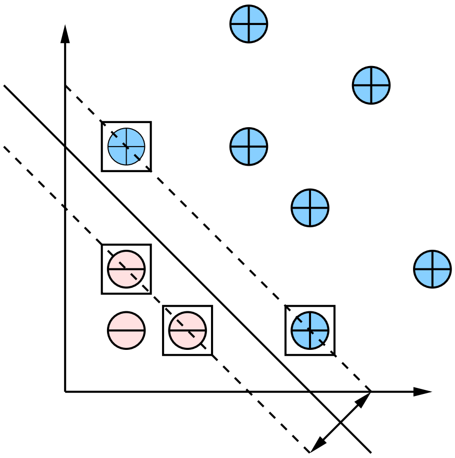


Figure 3.2: An example of a linearly separable training set in a two dimensional space (reproduced from [37]). The positive class examples are shown as circled plusses, whereas the negative class examples as circled minuses. The support vectors are enclosed in squares. The optimal hyperplane is shown by the solid line. The margin is shown by the two-headed arrow.

tion, i.e., $\xi_r > 0$ only if $\mathbf{o}_r$ violates the constraint in equation (3.29) otherwise $\xi_r = 0$ [37]. The new constraints may be expressed as

$$y_r(\mathbf{m} \cdot \mathbf{o}_r + b) \geq 1 - \xi_r, \quad \text{for all} \;\; r \tag{3.34}$$

The sum of the slack variables gives an upper bound on the number of misclassified training examples [25]. Minimising it one finds some minimal subset of vectors which if excluded yields a linearly separable training set [37]. This may be expressed formally as the following constrained optimisation problem

$$\{\widehat{\mathbf{m}}, \widehat{b}\} = \arg\min_{\mathbf{m},b} \left\{ \frac{1}{2} \|\mathbf{m}\|_2^2 + C \sum_{r=1}^{R} \xi_r \right\} \tag{3.35}$$

subject to the constraints in equation (3.34) and non-negativity of the slack variables [37]. The larger is the constant C, the more penalty is given to the training vectors violating the margin [25]. The solution to this optimisation problem is called the *soft margin hyperplane* [37]. For optimisation, the following *dual* [21] constrained optimisation problem is usually solved

$$\widehat{\boldsymbol{\alpha}}^{\texttt{svm}} = \arg\max_{\boldsymbol{\alpha}^{\texttt{svm}}} \left\{ \sum_{r=1}^{R} \alpha_r^{\texttt{svm}} - \frac{1}{2} \sum_{r=1}^{R} \sum_{r'=1}^{R} \alpha_r^{\texttt{svm}} \alpha_{r'}^{\texttt{svm}} y_r y_{r'} \mathbf{o}_r \cdot \mathbf{o}_{r'} \right\} \tag{3.36}$$

subject to

$$\sum_{r=1}^{R} \alpha_r^{\texttt{svm}} y_r = 0 \quad \text{and} \quad 0 \leq \alpha_r^{\texttt{svm}} \leq C, \quad \text{for all} \;\; r \tag{3.37}$$

where $\alpha_1^{\texttt{svm}}, \ldots, \alpha_R^{\texttt{svm}}$ are Lagrange multipliers [37]. At the optimality, it is possible to express the weight vector $\mathbf{m}$ in equation (3.33) as the following *sparse* linear combination of training vectors [37]

$$\mathbf{m} = \sum_{r=1}^{R} \alpha_r^{\texttt{svm}} y_r \mathbf{o}_r \tag{3.38}$$

where $\alpha_r^{\texttt{svm}} > 0$ only if $\mathbf{o}_r$ is the support vector. The decision function in equation (3.33) then may be expressed as

$$\widehat{y} = \operatorname{sgn}\left(\sum_{r=1}^{R} \alpha_r^{\texttt{svm}} y_r \mathbf{o}_r \cdot \mathbf{o} + b \right) \tag{3.39}$$

where, due to sparsity of the Lagrange multipliers, the summation may be performed only over the support vectors [37].

So far only linear decision surfaces have been considered. In order to extend the above approach to nonlinear decision surfaces, *support vector machines* (SVM) have been proposed [244]. Through some nonlinear *mapping* $\boldsymbol{\phi}(\cdot)$ chosen a priori, the SVMs map original vectors into a high-dimensional feature space $\boldsymbol{\Phi}$ where an optimal/soft margin hyperplane is constructed [244]. The dual constrained optimisation problem in equation (3.36) subject to the same constraints may be expressed in the new space as

$$\widehat{\boldsymbol{\alpha}}^{\texttt{svm}} = \arg\max_{\boldsymbol{\alpha}^{\texttt{svm}}} \left\{ \sum_{r=1}^{R} \alpha_r^{\texttt{svm}} - \frac{1}{2} \sum_{r=1}^{R} \sum_{r'=1}^{R} \alpha_r^{\texttt{svm}} \alpha_{r'}^{\texttt{svm}} y_r y_{r'} \boldsymbol{\phi}(\mathbf{o}_r) \cdot \boldsymbol{\phi}(\mathbf{o}_{r'}) \right\} \tag{3.40}$$

where $\boldsymbol{\phi}(\mathbf{o}_r)$ is the mapped version of the original vector $\mathbf{o}_r$ [19]. The decision function in equation (3.39) may be expressed in the new space as [19, 37]

$$\widehat{y} = \operatorname{sgn}\left(\sum_{r=1}^{R} \alpha_r^{\texttt{svm}} y_r \boldsymbol{\phi}(\mathbf{o}_r) \cdot \boldsymbol{\phi}(\mathbf{o}) + b\right) \tag{3.41}$$

For instance, when two-dimensional vectors are mapped according to $\boldsymbol{\phi}(\mathbf{o}) = [\ o_1^2 \ \ \sqrt{2}o_1o_2 \ \ o_2^2\ ]^{\mathsf{T}}$, an optimal/soft margin hyperplane constructed in the new space yields a nonlinear, degree 2 polynomial, decision surface in the original space [25].

As the dimensionality of feature vectors grows the SVM approach, as outlined above, may become computationally infeasible [37, 244]. In order to address this issue, a *kernel trick* [3] may be applied [19]. The kernel trick makes use of a *kernel function* $k(\cdot, \cdot)$ [95] to compute dot-products, such as in equation (3.40), *implicitly* without mapping the original vectors [244]

$$\boldsymbol{\phi}(\mathbf{o}_r) \cdot \boldsymbol{\phi}(\mathbf{o}_{r'}) = k(\mathbf{o}_r, \mathbf{o}_{r'}) \tag{3.42}$$

By using different kernels it is possible to construct different SVMs with arbitrary types of decision surfaces by solving the following dual constrained optimisation problem subject to the same constraints as the problem in equation (3.36) [37]

$$\widehat{\boldsymbol{\alpha}}^{\texttt{svm}} = \arg\max_{\boldsymbol{\alpha}^{\texttt{svm}}} \left\{ \sum_{r=1}^{R} \alpha_r^{\texttt{svm}} - \frac{1}{2} \sum_{r=1}^{R} \sum_{r'=1}^{R} \alpha_r^{\texttt{svm}} \alpha_{r'}^{\texttt{svm}} y_r y_{r'} k(\mathbf{o}_r, \mathbf{o}_{r'}) \right\} \tag{3.43}$$

The decision function of these SVMs has the following form [244]

$$\widehat{y} = \operatorname{sgn}\left(\sum_{r=1}^{R} \alpha_r^{\texttt{svm}} y_r k(\mathbf{o}_r, \mathbf{o}) + b\right) \tag{3.44}$$

There are several choices of kernel to use such as those given by Table 3.1. In particular, when the mapped vectors are defined by $\boldsymbol{\phi}(\mathbf{o}) = [\ o_1^2 \ \ \sqrt{2}o_1o_2 \ \ o_2^2\ ]^{\mathsf{T}}$ the dot-product can be implicitly computed by the degree $p = 2$ *homogeneous polynomial kernel* [25] listed in Table 3.1.

Type	Form $k(\mathbf{o}, \mathbf{o}')$	Parameters
Linear	$\mathbf{o} \cdot \mathbf{o}'$	–
Homogeneous polynomial	$(\mathbf{o} \cdot \mathbf{o}')^p$	p
Inhomogeneous polynomial	$(a + \mathbf{o} \cdot \mathbf{o}')^p$	a, p
Laplacian	$\exp(-\frac{1}{\sigma}\|\mathbf{o} - \mathbf{o}'\|_2)$	σ
RBF	$\exp(-\frac{1}{2\sigma^2}\|\mathbf{o} - \mathbf{o}'\|_2^2)$	σ

Table 3.1: Examples of kernels for fixed-length data

3.2.2 Dynamic kernels

So far it has been assumed that the data has a form of fixed-length vectors. As was discussed in Chapter 2, the speech data typically has a form of variable-length sequences. In order to address this inconsistency, *dynamic kernels* have been developed [101, 128, 145, 216]. These kernels typically map variable-length sequences into a fixed-dimensional feature space where an inner product can be calculated [145, 217]. Given a pair of observation sequences $\mathbf{O}_{1:T}$ and $\mathbf{O}'_{1:T'}$, the dynamic kernel may be defined by [101]

$$k(\mathbf{O}_{1:T}, \mathbf{O}'_{1:T'}) = \boldsymbol{\phi}(\mathbf{O}_{1:T})^{\mathsf{T}} \boldsymbol{\Sigma}_{\Phi}^{-1} \boldsymbol{\phi}(\mathbf{O}'_{1:T'}) \tag{3.45}$$

where $\boldsymbol{\phi}(\cdot)$ are feature-functions, $\boldsymbol{\Sigma}_{\Phi}$ is a *metric*, which defines distance in the feature space [101]. The simplest example of feature-functions was given with MaxEnt models in equation (3.12). A number of alternative approaches will be discussed in Chapter 6. The choice of metric $\boldsymbol{\Sigma}_{\Phi}$ in equation (3.45) is often not clear [145]. As the SVMs are sensitive to data scaling [214], it is advantageous to adopt a maximally non-committal metric [216]. One such metric is given by

$$\boldsymbol{\Sigma}_{\Phi} = \mathcal{E}\{(\boldsymbol{\phi}(\mathbf{O}_{1:T}) - \boldsymbol{\mu}_{\Phi})(\boldsymbol{\phi}(\mathbf{O}_{1:T}) - \boldsymbol{\mu}_{\Phi})^{\mathsf{T}}\} \tag{3.46}$$

where expectation, $\mathcal{E}\{\cdot\}$, is taken over all possible observation sequences and $\boldsymbol{\mu}_{\Phi} = \mathcal{E}\{\boldsymbol{\phi}(\mathbf{O}_{1:T})\}$ [216]. For some feature-functions these expectations may not have closed-form solutions [128]. In order to address this issue, the use of empirical estimate may be considered [43, 68]

$$\mathbf{\Sigma}_{\Phi} = \frac{1}{R}\sum_{r=1}^{R}(\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}) - \boldsymbol{\mu}_{\Phi})(\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}) - \boldsymbol{\mu}_{\Phi})^{\mathsf{T}} \tag{3.47}$$

where $\boldsymbol{\mu}_{\Phi} = \frac{1}{R}\sum_{r=1}^{R}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)})$. For high-dimensional score-spaces computing and storing $\mathbf{\Sigma}_{\Phi}$ based on equation (3.47) can be computationally expensive [128]. In order to address this issue, further approximations may be applied, such as diagonal approximation, $\text{diag}(\mathbf{\Sigma}_{\Phi})$, [217] which provides a reasonable approximation to $\mathbf{\Sigma}_{\Phi}$ in equation (3.47) whilst reducing the computational cost associated with inverting a full matrix to inverting a diagonal matrix [128].

Given dynamic kernel, the SVM can be constructed on variable-length sequence data by solving the dual constrained optimisation problem [101]

$$\widehat{\boldsymbol{\alpha}}^{\texttt{svm}} = \operatorname*{arg\,max}_{\boldsymbol{\alpha}^{\texttt{svm}}}\left\{\sum_{r=1}^{R}\alpha_r^{\texttt{svm}} - \frac{1}{2}\sum_{r=1}^{R}\sum_{r'=1}^{R'}\alpha_r^{\texttt{svm}}\alpha_{r'}^{\texttt{svm}}y_r y_{r'}k(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{O}_{1:T_{r'}}^{(r')})\right\} \tag{3.48}$$

subject to the same constraints as equation (3.43). The decision function with these SVMs has the following form [72, 101]

$$\widehat{y} = \text{sgn}\left(\sum_{r=1}^{R}\alpha_r^{\texttt{svm}}y_r k(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{O}_{1:T}) + b\right) \tag{3.49}$$

3.2.3 Multi-class extensions

So far the SVM have been assumed to be a binary classifier. There are options to extend the SVM to handle multiple classes [38]. Section 3.2.3.1 discusses the first option which is to reduce the multi-class problem into multiple binary problems [20, 126]. Section 3.2.3.2 then discusses the second option which is to modify the SVM classification and training to handle multiple classes [38, 244, 255].

3.2.3.1 One-versus-one classifiers

There are a number of options for using voting schemes with the SVMs for multi-class classification [72]. The schemes differ in the number of SVMs, the amount of training data and the number of classifications.

One of them is a *one-versus-the rest* (1-v-rest) classifier [20, 244]. Here, the SVMs are constructed for each of K classes such that the correct class provides with the positive class examples and the rest classes provide with the negative class examples [244]. Classification of a test observation sequence may be performed using a *winner-takes-all* strategy in which a class with the largest argument of decision function is selected [97].

Another one is a *one-versus-one* (1-v-1) classifier [57, 126]. Here, the SVMs are constructed for each pair of the classes; for a K-class problem, a total of $\frac{1}{2}K(K-1)$ SVMs is constructed [97]. Classification of test observation sequences can be performed using a *max-wins* strategy [57] in which a class with the largest number of pairwise ”wins” is selected with ties between classes broken randomly or resolved by an alternative, for example, generative, classifier [68].

3.2.3.2 Multi-class SVMs

An alternative to combining multiple binary SVMs is to construct a direct multi-class SVM by solving a single optimisation problem [196]. As an illustration, this section discusses multi-class SVMs where decision functions are given by

$$\widehat{\omega} = \arg\max_{\omega} \left\{ \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{1:T}, \omega) \right\} \tag{3.50}$$

where $\boldsymbol{\alpha}$ are parameters and $\boldsymbol{\phi}(\cdot)$ is a feature-function and ω is a sentence class label. Note that the same form of decision function was used for inference with MaxEnt models (see Section 3.1). There are a number of feature-functions that can be adopted with this form [73, 278]. The simplest example is to adopt the MaxEnt feature-function in equation (3.12). Given a feature-function, there are a number of approaches for training with this form [38, 244, 255]. Common to these approaches is that a single constrained optimisation problem with a quadratic objective function may be formulated [196]. The approaches differ in the form of constraints to be satisfied.

One of them [244, 255] requires dot-products for each reference sentence label to be at least by one larger than they are for *every* competing sentence label. By introducing slack variables to handle training sets violating this requirement, the constraints may be expressed as [120]

$$\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega_r) + \delta(\omega, \omega_r) - \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega) \geq 1 - \xi_{r,\omega} \tag{3.51}$$

This results in $K-1$ constraints per each training example [196].

Another approach [38] requires dot-products for each reference sentence label to be at least by one larger than they are for the *largest* competing sentence label. Note that all other competing sentence labels would have smaller dot-products and hence automatically satisfy this requirement. By introducing slack variables to handle training sets violating the above requirement, the constraints may be expressed as [38]

$$\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega_r) - \max_{\omega\neq\omega_r}\Big\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega) - \delta(\omega, \omega_r)\Big\} \geq 1 - \xi_r \tag{3.52}$$

This results in 1 rather than $K-1$ constraints per each training example [196].

An interesting aspect of the last approach is that it may be shown [277, 278] to be related to large margin training of maximum entropy models discussed in Section 3.1.2.3. In order to verify this, consider converting the associated constrained optimisation problem into an unconstrained optimisation problem. The constrained optimisation problem may be expressed as [120, 196]

$$\{\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\xi}}\} = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\xi}}\left\{\frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 + C\sum_{r=1}^{R}\xi_r\right\} \tag{3.53}$$

subject to the constraints in equation (3.52). The slack variables above may be expressed as [97][1]

$$\xi_r = \left[\max_{\omega\neq\omega_r}\Big\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega) + 1 - \delta(\omega, \omega_r)\Big\} - \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \omega_r)\right]_+ \tag{3.54}$$

Subsituting equation (3.54) into equation (3.53) yields an *unconstrained* optimi-

[1] In the referred work, maximisation inside the hinge-loss function $[\,\cdot\,]_+$ is performed over all (as in equation 3.54) rather than only competing (as in equation 3.52) sentence labels which, however, does not change the final result.

sation problem of the form given by equation (3.26), where

$$\mathcal{F}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 + \\ C\sum_{r=1}^{R}\left[\max_{\omega\neq\omega_r}\left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\omega)+1-\delta(\omega,\omega_r)\right\}-\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\omega_r)\right]_+ \tag{3.55}$$

Adopting $\mathcal{L}(\omega,\omega_r) = 1-\delta(\omega,\omega_r)$ and $\sigma^{\mathsf{p}} = CR$ in equation (3.26) yields the same form of objective function as in equation (3.55).

3.2.4 Adaptation to speaker and noise

In order to work reliably in real-world applications any speech recognition system must be designed to be robust to changes in speaker and noise conditions [75]. For HMMs, an overview of adaptation to the speaker and noise conditions was given in Section 2.8. There has been some previous work on adapting the SVMs to speaker and noise conditions [71, 98, 139].

The use of *regularised adaptation* has been considered in [139]. It makes use of the existing unadapted SVM parameters to form a prior on the new adapted parameters subject to estimation on the adaptation data. The regularised adaptation is a general adaptation scheme, similar to maximum-a-posteriori adaptation [138], which makes no assumption about the nature of feature-functions [73]. However, when the feature-functions exhibit some structure the use of other adaptation schemes may be more advantageous [73]. Furthermore, when the amount of adaptation data is very limited, for example a single utterance, this scheme may be unfeasible [68].

Alternatively, the use of *sample selection de-biasing*[1] has been investigated in [98]. It yields resampling weights for the training data samples to "match" the adaptation data samples [68]. However, for some speech recognition problems, such as noise-robust speech recognition and rapid speaker adaptation, it is not possible to ensure that the target speaker or environment is well covered by the training data [68].

[1] In the machine learning literature (c. f. [98]) the sample selection bias refers to the situation when the distributions assumed to identically and independently draw training and testing samples do not match.

Finally, the feature-functions themselves can be modified to make them robust to the speaker and noise conditions [73]. The SVM then can be trained speaker and noise independent using those modified feature-functions similar to the MaxEnt. This approach will be discussed in Chapter 6.

3.3 Summary

This chapter has discussed two discriminative classifiers that have been applied to speech recognition tasks: maximum entropy models (MaxEnt) and support vector machines (SVM).

The MaxEnt classifier models posterior probabilities of sentence labels given observation sequences. The resulting form of posterior probability is a member of exponential family defined in terms of (natural) parameters and sufficient statistics. The MaxEnt relies on feature functions to extract statistics from sentence labels and variable-length observation sequences. In particular, this chapter discussed basic requirements to and gave several examples of feature functions that could be used. In addition to the standard conditional maximum likelihood training, alternative discriminative training criteria such as minimum Bayes risk and large margin were discussed. Adaptation to mismatches between training and test condition were also discussed.

The SVM classifier in the simplest, binary case, maps observation sequences into one of two sentence labels depending on which side of a hyperplane it falls. In particular, it discussed learning hyperplanes with maximal margins for linearly separable training sets, which may be expected to yield low generalisation errors, and soft margins, which are useful in situations of linearly inseparable training sets. In order to construct non-linear classifiers, the use of non-linear mappings and kernels were discussed. In order to address variable-length nature of observation sequences, the use of dynamic kernels was discussed. In order to enable multi-class classification, a range of extensions, such as reduction schemes and multi-class SVMs, were discussed. Adaptation to mismatches between training and test condition were also discussed.

Chapter 4

Extended acoustic code-breaking

The discriminative models discussed in Chapter 3 were presented as whole-sentence models where parameters are associated with individual sentences. Although for some tasks it may be a feasible option [168, 217], as the number of sentence classes increases such approach quickly becomes impractical [73]. There are several options to address this issue. One option is to introduce a structure into the discriminative model by breaking down sentence labels into sub-sentence units, such as words or phones, similar to the standard approach applied with the acoustic and language model of the generative classifier in Chapter 2. This option will be discussed in Chapter 5. This chapter discusses an alternative option which is to decompose the whole sentence recognition problem into a sequence of independent, typically, word classification sub-problems using acoustic code-breaking schemes [68, 128, 246, 247]. These sub-problems are then addressed by using whole-word models where parameters are associates with individual words.

In order to construct whole-word models, it is necessary to have sufficient training data for each word. Although this is possible for some tasks [68], for others, such as city name recognition, it is unlikely that there will be sufficient data. This is a known limitation of acoustic code-breaking schemes and means the schemes can not be applied to general problems [72]. This chapter proposes an *extended acoustic code-breaking* which addresses the above limitation by artificially generating the required data. As this artificial data can be generated for any word, the use of acoustic code-breaking can be examined in tasks that were previously not possible.

The rest of this chapter is organised as follows. Section 4.1 discusses acoustic code-breaking schemes. The extended acoustic code-breaking will be presented in Section 4.2. A summary of this chapter is given in Section 4.3.

4.1 Acoustic code-breaking

The acoustic code-breaking is a re-scoring approach to speech recognition in which the whole-sentence recognition problem is transformed into multiple sequential, independent, word classification sub-problems [247]. This provides a general framework for incorporating models that may not be possible to apply to continuous speech recognition tasks [247]. For example, if the sub-problems are limited to word-pairs then SVMs (Section 3.2) may be directly used.

A number of acoustic code-breaking schemes have been proposed [68, 128, 247]. Common to these schemes is the use of existing HMM-based speech recognition system to yield an initial representation, such as the 1-best hypothesis or word lattice (Section 2.6). Given this initial representation, these schemes attempt to isolate and characterise the regions of acoustic confusability to which the whole word models are applied [247]. This section discusses two such schemes.

Given an observation sequence, the variant of acoustic code-breaking in [68] illustrated in Figure 4.1 makes use of existing HMM-based speech recognition system to produce a 1-best hypothesis with segmentation. Using the time stamp information provided by the segmentation, the observation sequence is segmented into sub-sequences. Discriminative models can then be applied to classify each

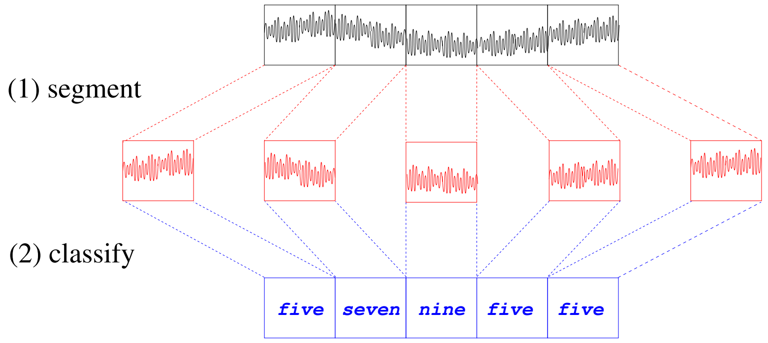
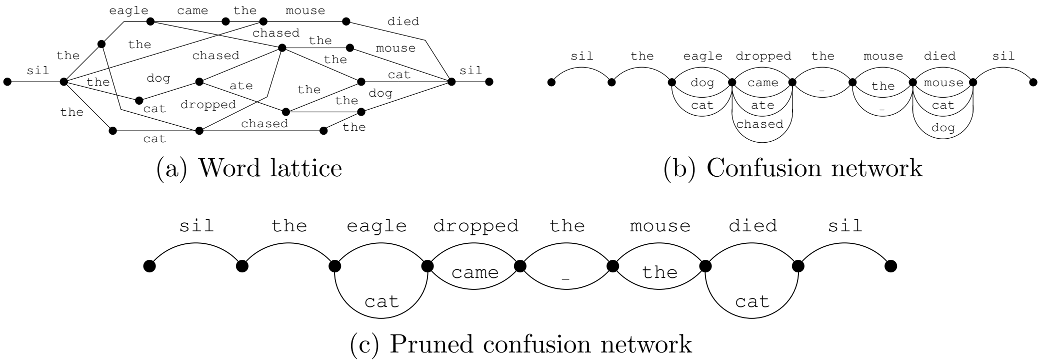


Figure 4.1: A simple form of acoustic code-breaking

sub-sequence into one of all possible word label classes. This variant of acoustic code-breaking was applied to digit string recognition tasks [72, 183], where the vocabulary of word label classes included digits from `oh` to `nine`, `zero` and silence. Examples of discriminative models examined included SVMs [68, 72], where multi-class decisions were made using the max-wins strategy (Section 3.2.3.1), multi-class SVMs [278], which as discussed in Section 3.2.3.2 are related to large margin trained maximum entropy models (Section 3.1.2.3).

In contrast, the variant of acoustic code-breaking in [128], similar to the schemes in [246, 247], was originally proposed for binary, word-pair re-scoring in a conversational telephone speech recognition task [51] using SVMs and generative augmented models [128, 130, 216]. This variant is illustrated in Figure 4.2 and may be described in three steps [128]. The first step makes use of an existing HMM-based speech recognition system to produce a word lattice, such as the one shown in Figure 4.2a. The second step converts this word lattice into a confusion network (Section 2.6), such as the one shown in Figure 4.2b. The third step prunes this confusion network so that each set of parallel arcs contains at most two parallel arcs as shown in Figure 4.2c. Those sets of two parallel arcs form confusable pairs. The observation sub-sequence for each confusable pair is extracted from the earliest start time to the latest end time of the two parallel arcs [128]. The discriminative models are then applied to re-score each confusable pair.



(a) Word lattice

(b) Confusion network

(c) Pruned confusion network

Figure 4.2: Confusion network based acoustic code-breaking

The presentation of acoustic code-breaking schemes have so far assumed the availability of whole-word models. Although for some tasks, such as digit string recognition, there usually is sufficient training data to estimate the model parameters, for others, such as conversational speech recognition, it is unlikely that there will be sufficient data [72]. Thus, the experiments in [128, 246] were limited to re-scoring only the most frequently occurring word-pair confusions, such as `can/can't`, `know/no` and `and/in` in [128]. This is a known limitation of acoustic code-breaking schemes and means the schemes can not be applied to tasks, such as city-name recognition, where there is limited, or no, examples of the words in the training data [72].

4.2 Extended acoustic code-breaking

In order to extend the acoustic code-breaking schemes to a more general setting it is important to find a solution to how to handle tasks where there is limited, or no, examples of possible words in the training data. One option would be to alter the level at which the discriminative models operate, for instance, a phone, similar to the use of phone units by HMM-based speech recognition systems in large vocabulary tasks (Section 2.3.1). However, this significantly complicates the issue of how to determine phone boundaries, which are much harder to estimate than word boundaries [72]. In addition, this approach is even more sensitive to the precise phone sequence being considered [246]. Another option, and the one proposed in this section, is to *artificially generate* examples of the words with limited, or no, representation in the training data. Thus, the extended acoustic code-breaking can be applied, for instance, to the conversational telephone speech recognition tasks in [128, 246] to re-score *all* rather than a small set of confusable pairs. It is also possible to consider other than binary confusions, for instance, by classifying each observation sub-sequence into one of all possible words.

In order to implement extended acoustic code-breaking, a restricted form of speech synthesis, which generates observation sequences, not waveforms, is effectively required. Thus, many of the issues commonly associated in speech synthesis with waveform generation, such as excitation and prosody [166], are not relevant to this approach. In order to generate observation sequences it is possible to use, for instance, concatenative or HMM-based speech synthesis [166]. As the acoustic code-breaking schemes rely on existing HMM-based speech recognition systems, the application of HMM-based speech synthesis will be considered. One additional advantage of this approach is that observation sequences with particular speaker and noise characteristics can be simply obtained by applying model-based adaptation/compensation approaches (Section 2.8) to modify HMM parameters prior to synthesis.

4.2.1 HMM synthesis

The simplest approach to synthesis with HMMs is to directly use models to generate observation sequences. Given word sequence $\mathbf{w}_{1:L}$, the observation sequence that maximises the likelihood would be generated by solving [234, 235]

$$\begin{aligned}\widehat{\mathbf{O}}_{1:T} &= \arg\max_{\mathbf{O}_{1:T}}\{p(\mathbf{O}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})\} & (4.1)\\ &= \arg\max_{\mathbf{O}_{1:T}}\left\{\sum_{\mathbf{q}} P(\mathbf{q}|\mathbf{w}_{1:L};\boldsymbol{\lambda})p(\mathbf{O}_{1:T}|\mathbf{q},\mathbf{w}_{1:L};\boldsymbol{\lambda})\right\} & (4.2)\end{aligned}$$

There is no known way to analytically solve this problem [269]. A commonly adopted approach is to use Viterbi approximation [272]. First, the optimal state-component sequence that maximises the probability of state-component sequences is found

$$\widehat{\mathbf{q}}_{1:T} = \arg\max_{\mathbf{q}_{1:T}}\{P(\mathbf{q}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})\} \tag{4.3}$$

Second, the observation sequence that maximises the likelihood given the solution to the first maximisation problem in equation (4.3) is solved for

$$\widehat{\mathbf{O}}_{1:T} = \arg\max_{\mathbf{O}_{1:T}}\{p(\mathbf{O}_{1:T}|\widehat{\mathbf{q}}_{1:T},\mathbf{w}_{1:L};\boldsymbol{\lambda})\} \tag{4.4}$$

Equation (4.3) is usually solved based on explicit HMM state duration densities [269]. The simplest option is to adopt the inherent HMM state duration density in equation (2.11) [233]. Alternatively, it is possible to consider Gaussian distributions whose parameters may be estimated based on additional statistics accumulated in ML estimation [262].

The solution to the problem in equation (4.4) is simplified if the distribution of observation sequences given a state-component sequence $\mathbf{q}_{1:T}$ is Gaussian [269, 274]

$$p(\mathbf{O}_{1:T}|\mathbf{q}_{1:T}, \mathbf{w}_{1:L}; \boldsymbol{\lambda}) = \mathcal{N}(\mathbf{O}_{1:T}; \boldsymbol{\mu}_{\mathbf{q}_{1:T}}, \boldsymbol{\Sigma}_{\mathbf{q}_{1:T}}) \tag{4.5}$$

where the mean vector $\boldsymbol{\mu}_{\mathbf{q}_{1:T}}$ and the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{q}_{1:T}}$ are given by

$$\boldsymbol{\mu}_{\mathbf{q}_{1:T}} = \begin{bmatrix} \boldsymbol{\mu}_{q_1} \\ \vdots \\ \boldsymbol{\mu}_{q_T} \end{bmatrix}, \qquad \boldsymbol{\Sigma}_{\mathbf{q}_{1:T}} = \begin{bmatrix} \boldsymbol{\Sigma}_{q_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{\Sigma}_{q_T} \end{bmatrix} \tag{4.6}$$

Thus, given the solution to the first problem, $\widehat{\mathbf{q}}_{1:T}$, the solution to the second problem is given by the associated mean [232]

$$\widehat{\mathbf{O}}_{1:T} = \boldsymbol{\mu}_{\widehat{\mathbf{q}}_{1:T}} \tag{4.7}$$

The procedure above is a simple generative process but the generated observation sequence will be based on the same conditional independence assumptions as the underlying HMM [72]. In particular, the synthesised observation sequence is obtained as a sequence of HMM state-component mean vectors - a *piece-wise stationary trajectory* - causing discontinuity on transitioning from one HMM state-component mean vector to another [269]. This has been found to result in "clicks" in the reconstructed speech signal, degrading the naturalness [152].

4.2.2 Statistical HMM synthesis

In order to overcome conditional independence assumptions that are often cited as an issue with the HMM synthesis discussed in Section 4.2.1 [233, 234, 274], it is possible to apply statistical HMM synthesis [235]. The idea behind statistical HMM synthesis is to synthesise observation sequences by taking into account the deterministic relationship that exists between the static and dynamic parts of observation vectors (Section 2.1.2). For instance,

$$\begin{bmatrix} \mathbf{o}_{t-1} \\ \mathbf{o}_t \\ \mathbf{o}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{o}^{\mathsf{s}}_{t-1} \\ \Delta^{(1)}\mathbf{o}_{t-1} \\ \mathbf{o}^{\mathsf{s}}_{t} \\ \Delta^{(1)}\mathbf{o}_{t} \\ \mathbf{o}^{\mathsf{s}}_{t+1} \\ \Delta^{(1)}\mathbf{o}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\frac{\mathbf{I}}{2} & \mathbf{0} & \frac{\mathbf{I}}{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{\mathbf{I}}{2} & \mathbf{0} & \frac{\mathbf{I}}{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{\mathbf{I}}{2} & \mathbf{0} & \frac{\mathbf{I}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{o}^{\mathsf{s}}_{t-2} \\ \mathbf{o}^{\mathsf{s}}_{t-1} \\ \mathbf{o}^{\mathsf{s}}_{t} \\ \mathbf{o}^{\mathsf{s}}_{t+1} \\ \mathbf{o}^{\mathsf{s}}_{t+2} \end{bmatrix} \tag{4.8}$$

expresses the relationship between static $\mathbf{o}^{\mathsf{s}}_{t-2}, \ldots, \mathbf{o}^{\mathsf{s}}_{t+2}$ and complete $\mathbf{o}_{t-1}$, $\mathbf{o}_t$, $\mathbf{o}_{t+1}$ observations assuming that the dynamic observations $\Delta^{(1)}\mathbf{o}_{t-1}$, $\Delta^{(1)}\mathbf{o}_{t}$, $\Delta^{(1)}\mathbf{o}_{t+1}$ are obtained as simple differences according to equation (2.4). It is also possible to express the relationship between complete $\mathbf{O}_{1:T}$ and static $\mathbf{O}^{\mathsf{s}}_{1:T}$ sequences as

$$\mathbf{O}_{1:T} = \mathbf{A}\mathbf{O}^{\mathsf{s}}_{1:T} \tag{4.9}$$

where $\mathbf{A}$ is the window matrix [232]. For a given state-component sequence $\mathbf{q}_{1:T}$, the likelihood of static observation sequences may be computed by appropriately normalising the likelihood of complete observation sequences in equation (4.5) [269]

$$\frac{1}{Z}\mathcal{N}(\mathbf{A}\mathbf{O}^{\mathsf{s}}_{1:T}; \boldsymbol{\mu}_{\mathbf{q}_{1:T}}, \boldsymbol{\Sigma}_{\mathbf{q}_{1:T}}) = \mathcal{N}(\mathbf{O}^{\mathsf{s}}_{1:T}; \boldsymbol{\mu}^{\mathsf{s}}_{\mathbf{q}_{1:T}}, \boldsymbol{\Sigma}^{\mathsf{s}}_{\mathbf{q}_{1:T}}) \tag{4.10}$$

where Z is the normalisation term. The static mean vector $\boldsymbol{\mu}^{\mathsf{s}}_{\mathbf{q}_{1:T}}$ and the static covariance matrix $\boldsymbol{\Sigma}^{\mathsf{s}}_{\mathbf{q}_{1:T}}$ associated with the likelihood of static observation sequences may be found from the following relationships that exists between the static and standard means and covariances [232]

$$\begin{aligned} \boldsymbol{\Sigma}^{\mathsf{s}^{-1}}_{\mathbf{q}_{1:T}} &= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_{1:T}}\mathbf{A} && (4.11) \\ \boldsymbol{\Sigma}^{\mathsf{s}^{-1}}_{\mathbf{q}_{1:T}}\boldsymbol{\mu}^{\mathsf{s}}_{\mathbf{q}_{1:T}} &= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_{1:T}}\boldsymbol{\mu}_{\mathbf{q}_{1:T}} && (4.12) \end{aligned}$$

In contrast to the standard covariance $\boldsymbol{\Sigma}_{\mathbf{q}_{1:T}}$ matrix in equation (4.6), which is a block-diagonal matrix, the static covariance $\boldsymbol{\Sigma}^{\mathsf{s}}_{\mathbf{q}_{1:T}}$ does not have a block-diagonal structure since $\mathbf{A}$ is not block diagonal [75]. Solving for the static observation sequence that maximises the likelihood of static observation sequences given a state-component sequence $\widehat{\mathbf{q}}_{1:T}$ yields the associated static mean

$$\widehat{\mathbf{O}}^{\mathsf{s}}_{1:T} = \boldsymbol{\mu}^{\mathsf{s}}_{\widehat{\mathbf{q}}_{1:T}} = \left(\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\widehat{\mathbf{q}}_{1:T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{\widehat{\mathbf{q}}_{1:T}}\boldsymbol{\mu}_{\widehat{\mathbf{q}}_{1:T}} \tag{4.13}$$

as the underlying distribution is Gaussian [75]. The complete synthesised observation sequence, $\widehat{\mathbf{O}}_{1:T}$, then can be obtained according to equation (4.9).

Compared to the HMM synthesis, the statistical HMM synthesis has so far introduced only a few modifications yet the generated observation sequence already will not be based on the same conditional independence assumptions as the underlying HMM [269]. In particular, the trajectory of synthesised static observation sequence is no longer piece-wise stationary [75]. A number of further refinements have been proposed. Examples include replacing the Viterbi approximation to solve equation (4.1) by an EM algorithm [234] and using alternative, ML-based [269] and discriminative [260], estimation criteria to yield HMM parameters which when used to generate static observation sequences they are a "good" model of the training data [75], average voice models providing robust and steady examples with limited amounts of adaptation data , speech synthesis algorithms considering global variance which produce trajectories with dynamic range approaching real examples [231]. The first two of these refinements are discussed below.

Rather than adopting the Viterbi approximation to solve equation (4.1), an auxiliary function can be formulated to search for an observation sequence which maximises the likelihood [234]

$$\mathcal{Q}(\mathbf{O}_{1:T}, \widehat{\mathbf{O}}_{1:T}; \boldsymbol{\lambda}) = \sum_{\mathbf{q}_{1:T}} P(\mathbf{q}_{1:T}|\mathbf{O}_{1:T}, \mathbf{w}_{1:L}; \boldsymbol{\lambda}) \log(p(\widehat{\mathbf{O}}_{1:T}, \mathbf{q}_{1:T}|\mathbf{w}_{1:L}; \boldsymbol{\lambda})) \tag{4.14}$$

where $\mathbf{O}_{1:T}$ is the current and $\widehat{\mathbf{O}}_{1:T}$ is the new observation sequence. Taking derivative and solving with respect to the new static observation sequence $\widehat{\mathbf{O}}^{\mathsf{s}}_{1:T}$ yields [234]

$$\widehat{\mathbf{O}}^{\mathsf{s}}_{1:T} = \left(\mathbf{A}^{\mathsf{T}}\overline{\boldsymbol{\Sigma}^{-1}_{\widehat{\mathbf{q}}_{1:T}}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathsf{T}}\overline{\boldsymbol{\Sigma}^{-1}_{\widehat{\mathbf{q}}_{1:T}}\boldsymbol{\mu}_{\widehat{\mathbf{q}}_{1:T}}} \tag{4.15}$$

where

$$\overline{\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_{1:T}}} = \begin{bmatrix} \sum\limits_{\{j,m\}} \gamma_{j,m}(1)\boldsymbol{\Sigma}^{-1}_{j,m} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \sum\limits_{\{j,m\}} \gamma_{j,m}(T)\boldsymbol{\Sigma}^{-1}_{j,m} \end{bmatrix} \tag{4.16}$$

and

$$\overline{\boldsymbol{\Sigma}^{-1}_{\mathbf{q}_{1:T}}\boldsymbol{\mu}_{\mathbf{q}_{1:T}}} = \begin{bmatrix} \sum\limits_{\{j,m\}} \gamma_{j,m}(1)\boldsymbol{\Sigma}^{-1}_{j,m}\boldsymbol{\mu}_{j,m} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \sum\limits_{\{j,m\}} \gamma_{j,m}(T)\boldsymbol{\Sigma}^{-1}_{j,m}\boldsymbol{\mu}_{j,m} \end{bmatrix} \tag{4.17}$$

Compared to the static observation sequence in equation (4.13), the contribution of all state-components weighted by the respective state-component occupancies is taken into account.

The HMM parameters have so far been assumed to be estimated, for example, using the approaches discussed in Chapter 2. These approaches, however, do not take into account the explicit relationship between complete and static sequences. In order to address this issue, the HMM can be reformulate as a model of static observation sequences [236]. This is the basis of the *trajectory HMM* [269]

$$p(\mathbf{O}^{\mathsf{s}}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda}) = \sum_{\mathbf{q}} P(\mathbf{q}|\mathbf{w}_{1:L};\boldsymbol{\lambda})\mathcal{N}(\mathbf{O}^{\mathsf{s}}_{1:T};\boldsymbol{\mu}^{\mathsf{s}}_{\mathbf{q}_{1:T}},\boldsymbol{\Sigma}^{\mathsf{s}}_{\mathbf{q}_{1:T}}) \tag{4.18}$$

where the mean vector $\boldsymbol{\mu}^{\mathsf{s}}_{\mathbf{q}_{1:T}}$ and covariance matrix $\boldsymbol{\Sigma}^{\mathsf{s}}_{\mathbf{q}_{1:T}}$ are given by equations (4.12) and (4.11). The trajectory HMM can not directly use the standard Viterbi algorithm (Section 2.2.1) to obtain the optimal state sequence [75]. In order to address this issue, a frame delayed version has been proposed [273] which, however, is more expensive than the standard Viterbi algorithm [75]. The trajectory HMM parameters, the underlying HMM parameters $\boldsymbol{\lambda}$, can be estimated using ML [269] and discriminative [260] criteria.

4.3 Summary

The discriminative models in Chapter 3 were presented as the whole sentence models which associate parameters with individual sentences. As the number of sentence classes increases, the whole sentence modelling quickly becomes impractical. This chapter has discussed acoustic code-breaking schemes which alter the level at which the discriminative models operate, a word, and decompose the whole sentence recognition problem into sequential, independent, word classification sub-problems. The application of these schemes to tasks, where there is limited, or no, examples of the words in the training data, has so far been focused on re-scoring a small number of the most frequently occurring confusion pairs. This has limited possible gains from the use of these schemes.

In order to make acoustic code-breaking schemes more generally applied, this chapter has extended these schemes to handle situations where there is limited or no examples of the words in the training data. The approach proposed is to artificially generate data. Essentially, a constrained version of speech synthesis, which only generates observation sequences, not waveforms, is required. This chapter has focused on HMM-based speech synthesis. One advantage of this approach is that observation sequences with particular speaker and noise characteristics can be simply produced by adapting HMM to speaker and noise conditions using model-based techniques. There were two HMM-based speech synthesis approaches discussed. The first approach is a simple generative process that directly uses HMMs to generate observation sequences. The synthesised observation sequences will be based on the same conditional independence assumptions as the underlying HMMs, which yields piece-wise stationary trajectories. In order to overcome these assumptions, the second approach takes into account the deterministic relationship that exists between the static and dynamic parts of observation vectors to generate the ”optimal” static observation sequences. Although the distribution of static observation sequences is parametrised with HMM parameters, the same conditional independence assumptions are not present, which yields trajectories that are no longer piece-wise stationary. Approaches, such as trajectory HMMs, average voice models and speech synthesis considering global variance, can be also incorporated into extended acoustic code-breaking to improve quality of generated data.

Chapter 5

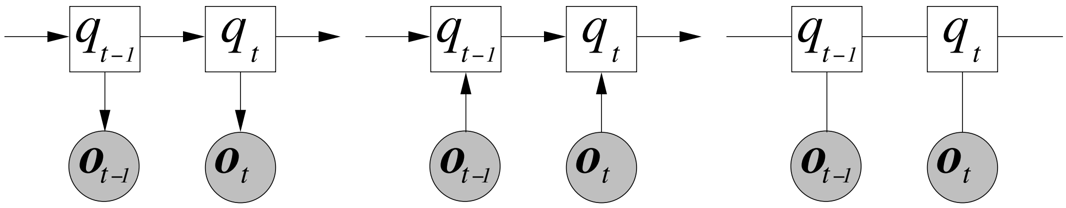
Structured discriminative models

The discriminative models discussed in Chapter 3 were presented as the whole sentence models that associate parameters with individual sentence labels. Although for some tasks it may be a feasible option [168, 217], as the number of sentence classes increases such approach quickly becomes impractical [73]. There are several options to address this issue. Chapter 4 discussed one option which is to associate parameters with individual words and make use of acoustic code-breaking schemes to decompose the whole sentence recognition problem into a sequence of independent word classification problems. This chapter discusses another option which is to introduce a structure into the discriminative model by breaking down sentence labels into sub-sentence units, such as words or phones, similar to the standard approach applied with the acoustic and language model of the generative classifier in Chapter 2. Related approaches, where the discriminative models adopt the same type of sub-sentence units, have also been developed [127, 128, 223, 281]. These discriminative models are usually called *structured discriminative models* [73]. This chapter discusses some forms of the structure that have been incorporated (Section 5.1), handling of latent variables which relate observations with the sub-sentence units (Section 5.2), parameter estimation (Section 5.3) and adaptation to speaker and noise conditions (Section 5.4).

5.1 Model structures

Structured discriminative models aim to adopt the same sub-sentence units as the acoustic model, the HMM (Section 2.3), and language model, the n-gram model (Section 2.5), of the generative classifier [73]. There are a number of structural forms that have been examined [127, 128, 223, 281].

The simplest way to introduce structure into the discriminative classifier is to make use of graphical models that are closely linked to the dynamic Bayesian network of the HMM repeated in Figure 5.1a for the ease of reference. Two such graphical models are shown in Figure 5.1. The first graphical model in Figure 5.1b reverses the direction of arrows describing the observation-state relationship compared to the HMM. Here, at time t the posterior probability of being in the current state q_t, in addition to the previous state q_{t-1}, also depends on the current observation $\mathbf{o}_t$ [85]. This serves the basis of maximum entropy Markov models (MEMM) [127, 153]. The second graphical model in Figure 5.1c is an undirected graph compared to the HMM and MEMM. Here, at time t the "probability" of being in the current state q_t given the previous state q_{t-1} and the current observation $\mathbf{o}_t$ is not normalised [85]. This serves the basis of a hidden conditional random field (HCRF) [85, 220, 222, 223]. The posterior probability associated with word sequence $\mathbf{w}_{1:L}$ in the HCRF may be expressed as [220]



(a) HMM (b) MEMM (c) HCRF

Figure 5.1: Dynamic Bayesian network for (a) hidden Markov model (HMM) and graphical models for frame-level structured models: (b) maximum entropy Markov model (MEMM), (c) hidden conditional random field (HCRF)

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T};\boldsymbol{\alpha}) = \frac{1}{Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})}\sum_{\mathbf{q}_{1:T}} \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\mathbf{q}_{1:T},\mathbf{w}_{1:L})) \tag{5.1}$$

where $Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})$ is a normalisation term, $\mathbf{q}_{1:T}$ is a state sequence in the composite sentence model for $\mathbf{w}_{1:L}$ (Section 2.3.2), discriminative model parameters $\boldsymbol{\alpha}$ and features $\boldsymbol{\phi}(\cdot)$ are given by

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_{\texttt{am}} \\ \boldsymbol{\alpha}_{\texttt{pm}} \\ \boldsymbol{\alpha}_{\texttt{lm}} \end{bmatrix}, \quad \boldsymbol{\phi}(\mathbf{O}_{1:T}, \mathbf{q}_{1:T}, \mathbf{w}_{1:L}) = \begin{bmatrix} \sum_{t=1}^{T} \boldsymbol{\phi}(\mathbf{o}_t, q_t) \\ \boldsymbol{\phi}(\mathbf{q}_{1:T}, \mathbf{w}_{1:L}) \\ \boldsymbol{\phi}(\mathbf{w}_{1:L}) \end{bmatrix} \tag{5.2}$$

where $\boldsymbol{\phi}(\mathbf{o}_t, q_t)$ are observation features, $\boldsymbol{\phi}(\mathbf{q}_{1:T}, \mathbf{w}_{1:L})$ and $\boldsymbol{\phi}(\mathbf{w}_{1:L})$ are suprasegmental features providing transition/pronunciation and language model features, $\boldsymbol{\alpha}_{\texttt{am}}$, $\boldsymbol{\alpha}_{\texttt{pm}}$ and $\boldsymbol{\alpha}_{\texttt{lm}}$ are the respective parameters.

The simplest example of observation features is given by [85]

$$\boldsymbol{\phi}(\mathbf{o}_t, q_t) = \begin{bmatrix} \vdots \\ \delta(q_t, S_j)\mathbf{o}_t \\ \delta(q_t, S_j)\text{vec}(\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}}) \\ \vdots \end{bmatrix} \tag{5.3}$$

where S_i and S_j are the HCRF states, vec$(\cdot)$ maps matrices into vectors by stacking columns on top of each other. For an N-state HCRF with d-dimensional observation vectors the number of observation features in equation (5.3) is $N(d+d^2)$. These features bear a resemblance to the HMM mean and covariance statistics in equations (2.49) and (2.50) (see Section 2.2.3) and may also be adopted with the MEMM [127]. Though simple, it is possible to show [90, 91] that HCRFs adopting these feature functions are related to discriminatively trained HMMs discussed in Section 2.7 [73]. In the following these features will be referred to as the *MEMM/HCRF features*.

The simplest option to define suprasegmental features is to set them to the standard HMM transition and n-gram model log-probabilities

$$\boldsymbol{\phi}(\mathbf{q}_{1:T}, \mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{q}_{1:T}|\mathbf{w}_{1:L}\boldsymbol{\lambda})\Big], \quad \boldsymbol{\phi}(\mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{w}_{1:L}))\Big] \tag{5.4}$$

where $P(\mathbf{q}_{1:T}|\mathbf{w}_{1:L};\boldsymbol{\lambda})$ is a probability of state sequence $\mathbf{q}_{1:T}$ in the composite HMM model for word sequence $\mathbf{w}_{1:L}$ and $P(\mathbf{w}_{1:L})$ is the probability of word sequence $\mathbf{w}_{1:L}$ given by the n-gram language model. Given these simple examples of observation (equation 5.3) and suprasegmental (equation 5.4) features the total dimensionality of features in equation (5.2) for an N-state HCRF would amount to $N(d + d^2) + 2$. Note that the number of parameters would also be equal to $N(d + d^2) + 2$. An overview of features possible to use with the HCRF will be given later in Chapter 6.

The use of MEMM/HCRF features implies that at each time the observation features depend on the current observation and state [73]. Although a fixed span of observations and states may be considered to relax this assumption [93, 225], the MEMM/HCRF will still generate T feature vectors for a T-length observation sequence [73]. An alternative option is to allow observations across a *word* or *phone segment* to contribute to the observation features [73]. This is the basis of segmental conditional random fields (SCRF) [281] and conditional augmented models (CAug) [128] respectively. The graphical model in Figure 5.2 illustrates the SCRF by showing two segments each comprising a word and observation subsequence given a particular segmentation. The posterior probability associated

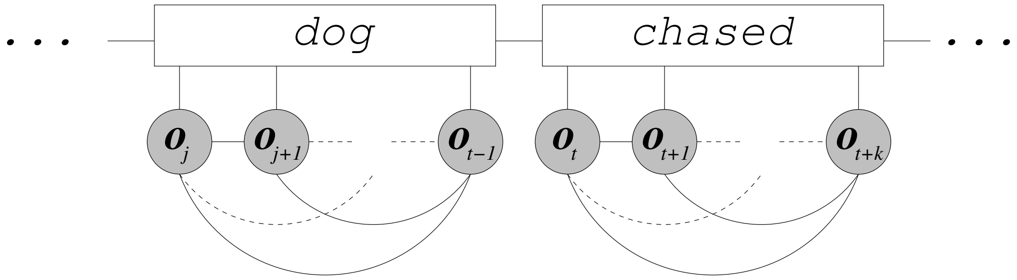


Figure 5.2: Graphical model for a word-level structured discriminative model given a particular segmentation (reproduced from [73])

with word sequence $\mathbf{w}_{1:L}$ may be expressed in SCRF/CAug as [73]

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T}; \boldsymbol{\alpha}) = \frac{1}{Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha})} \sum_{\mathbf{a}} \exp(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{1:T}, \mathbf{a}, \mathbf{w}_{1:L})) \tag{5.5}$$

where $Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha})$ is a normalisation term given by

$$Z(\mathbf{O}_{1:T}; \boldsymbol{\alpha}) = \sum_{\mathbf{w}} \sum_{\mathbf{a}} \exp(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{1:T}, \mathbf{a}, \mathbf{w})) \tag{5.6}$$

the discriminative model parameters $\boldsymbol{\alpha}$ have the same form as the HCRF parameters in equation (5.2) and features $\boldsymbol{\phi}(\cdot)$ are given by [73]

$$\boldsymbol{\phi}(\mathbf{O}_{1:T}, \mathbf{a}, \mathbf{w}_{1:L}) = \begin{bmatrix} \sum_{s=1}^{|\mathbf{a}|} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s) \\ \boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L}) \\ \boldsymbol{\phi}(\mathbf{w}_{1:L}) \end{bmatrix} \tag{5.7}$$

where a_s is a segment specifying segment identity a_s^{i}, a word in SCRF and phone in CAug, and observation sub-sequence $\mathbf{O}_{\{a_s\}}$ and $\mathbf{a} = \{a_1, \ldots, a_s, \ldots, a_{|\mathbf{a}|}\}$ is the segmentation, $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s)$ are acoustic segment features, $\boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L})$ and $\boldsymbol{\phi}(\mathbf{w}_{1:L})$ are suprasegmental features providing pronunciation and language model features. Note that acoustic segment features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s)$ are extracted from variable-length observation sub-sequences $\mathbf{O}_{\{a_s\}}$ unlike MEMM/HCRF observation features $\boldsymbol{\phi}(\mathbf{o}_t, q_t)$ in equation (5.3) which are extracted from the individual observations $\mathbf{o}_t$. One option to define acoustic segment features is to make use of frame-level features, in the same fashion as the MEMM/HCRF features [73]

$$\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s) = \sum_{t \in \{a_s\}} \boldsymbol{\phi}(\mathbf{o}_t, a_s) \tag{5.8}$$

The simplest option to define the suprasegmental features is to set them to the standard pronunciation $P(\mathbf{a}^{\text{i}}|\mathbf{w}_{1:L})$ and n-gram model $P(\mathbf{w}_{1:L})$ log-probabilities

$$\boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{a}^{\text{i}}|\mathbf{w}_{1:L}))\Big], \quad \boldsymbol{\phi}(\mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{w}_{1:L}))\Big] \tag{5.9}$$

The rest of this chapter assumes that the suprasegmental features are decomposable over individual words. Given these simple examples of observation (equation 5.8) and suprasegmental (equation 5.9) features, the total dimensionality of SCRF/CAug features in equation (5.7) would amount to $|\mathcal{V}|(d + d^2) + 2$ where $\mathcal{V}$ is a vocabulary of segment identities (a word in SCRF and a phone in CAug) and d is the dimensionality of observation vectors. An overview of features possible to use with the SCRF/CAug will be given later in Chapter 6.

5.2 Handling latent variables

The previous section has considered summing over latent variable sequences - hidden state sequences $\mathbf{q}_{1:T}$ with the MEMM/HCRF and segmentations $\mathbf{a}$ with the SCRF/CAug - to yield the posterior associated with word sequence $\mathbf{w}_{1:L}$. The use of direct summation over latent variable sequences becomes computationally expensive as the number of possible latent variable sequences increases [73]. An alternative option discussed in Section 5.2.1 is to devise recursions similar to the HMM forward-backward algorithm [73]. Another option discussed in Section 5.2.2 is to apply the equivalent of *Viterbi training and decoding* [251] by making use of a single latent variable sequence in the summation [73].

5.2.1 Forward-backward recursions

As noted earlier, the use of direct summation over hidden state sequences and segmentations becomes computationally excessive as the number of possible sequences increases. A related problem of summing over hidden state sequences with the HMM may be addressed by means of the forward-backward algorithm (Section 2.2.2). Related approaches have also been developed for the MEMM [153], HCRF [85, 224] and SCRF/CAug [281].

In the SCRF/CAug case, the forward $\alpha_a(t)$ and backward $\beta_a(t)$ quantities may be computed by

$$\alpha_a(t) = \sum_{a'}\sum_{\tau} \alpha_{a'}(\tau)\widetilde{P}(a|a')\exp\left(\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\tau+1:t},a)\right),\ \ \tau\in\{1,\ldots,t-1\} \tag{5.10}$$

$$\beta_a(t) = \sum_{a'}\sum_{\tau} \beta_{a'}(\tau)\widetilde{P}(a'|a)\exp\left(\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{t+1:\tau},a')\right),\ \ \tau\in\{t+1,\ldots,T\} \tag{5.11}$$

where $\widetilde{P}(a|a')$ is a segment transition score - the equivalent of HMM phone arc transition probability $P(a|a')$ - applying the pronunciation and language model scores. For example,

$$\widetilde{P}(a|a') = \exp\left(\begin{bmatrix}\alpha_{\mathtt{pm}}\\ \alpha_{\mathtt{lm}}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\log(P(a|w))\\ \log(P(w|w'))\end{bmatrix}\right) \tag{5.12}$$

may be used with SCRFs to apply standard pronunciation $P(a|w)$ and bigram $P(w|w')$ language model probabilities.[1] Compared to the standard forward-backward algorithm, there is an additional summation over a range of preceding/following time indices τ. The above recursions may be applied to compute the unnormalised posterior associated with word sequence $\mathbf{w}_{1:L}$ and the normalisation term $Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})$ in equation (5.5) by considering only those segments which are consistent with $\mathbf{w}_{1:L}$ and all possible segments respectively [281]. Under the assumption that feature extraction takes $\mathcal{O}(1)$ time, the computational complexity associated with these recursions is $\mathcal{O}(N^2T^2)$ [281] compared to $\mathcal{O}(N^2T)$ in the HMM case [49], where N is the number of segment identities/HMM states.

5.2.2 Viterbi training and decoding

As noted above, the alternative approach to summing over all hidden state sequences or segmentations is to perform the equivalent of Viterbi training and decoding [251]. This has been considered with the MEMM [127, 153], HCRF [163] and SCRF/CAug [128, 276, 281].

In the SCRF/CAug case, the posterior associated with word sequence $\mathbf{w}_{1:L}$ given segmentation $\widehat{\mathbf{a}}$ may be expressed as [73]

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T},\widehat{\mathbf{a}};\boldsymbol{\alpha}) = \frac{1}{Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\widehat{\mathbf{a}},\mathbf{w}_{1:L})) \tag{5.13}$$

where $Z(\mathbf{O}_{1:T};\boldsymbol{\alpha})$ is given by [128]

$$Z(\mathbf{O}_{1:T};\boldsymbol{\alpha}) = \sum_{\{\mathbf{w}',\widehat{\mathbf{a}}'\}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\widehat{\mathbf{a}}',\mathbf{w}')) \tag{5.14}$$

Note that summation in equation (5.14) is performed over all possible word sequences with associated given segmentations rather than all possible word sequences and segmentations as in equation (5.6).

In order to make use of this form, the segmentation must be provided for each word sequence. The simplest option is to obtain it from a generative classifier, such as the HMM [128]. However, the segmentation optimal with respect to

[1] It is assumed here that words w and w' are associated with segments a and a' respectively.

generative classifier may be sub-optimal with respect to discriminative classifier [276]. In order to derive the optimal with respect to SCRF/CAug segmentation, a semi-Markov variant [204] of the Viterbi algorithm may be applied [276]. The associated recursion may be expressed as [277, 281]

$$\phi_a(t) = \max_{a'} \max_{\tau} \left\{ \phi_{a'}(\tau) \widetilde{P}(a|a') \exp(\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\tau+1:t}, a)) \right\}, \quad \tau \in \{1, \ldots, t-1\} \tag{5.15}$$

where, compared to the Viterbi recursion in equation (2.22), there is an additional maximisation over a range of preceding time indices τ. Under the assumption that feature extraction takes $\mathcal{O}(1)$ time, the computational complexity associated with this recursion is $\mathcal{O}(N^2T^2)$ [281] compared to $\mathcal{O}(N^2T)$ in the HMM case [49], where N is the number of segment identities/HMM states.

5.3 Parameter estimation

In the same fashion as generative models, such as the HMM (Section 2.7), and standard, non-structured, discriminative models, such as MaxEnt (Section 3.1.2) it is possible to use a range of discriminative criteria with the structured discriminative models [73]. Examples include conditional maximum likelihood (CML), minimum word error (MWE)/minimum phone error (MPE) and large margin. The rest of this section will discuss optimisation of these criteria with the structured discriminative models in Sections 5.3.1, 5.3.2 and 5.3.3 respectively.

5.3.1 Optimisation of CML

The structured discriminative model parameter estimation based on the CML criterion may be performed by maximising the following objective function [73]

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \log(P(\mathbf{w}_{1:L_r}^{(r)} | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha})) \tag{5.16}$$

where $\boldsymbol{\alpha}$ are parameters and $\mathcal{D}$ is the supervised training data. The optimisation of this objective function is closely linked to the maximum mutual information (MMI) estimation of HMM parameters (Section 2.7.2.1), CML estimation of Max-

Ent parameters (Section 3.1.2.1) and have been examined with MEMM [127, 153], HCRF [85, 223] and SCRF/CAug [128, 281].

In the SCRF/CAug case, the posterior associated with word sequence $\mathbf{w}_{1:L}$ in equation (5.5) involves summing over all segmentations, compared to the MaxEnt. Using this form, the CML objective function in equation (3.7) may be expressed as [128, 169]

$$\mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\Bigg[\log\Bigg(\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w}^{(r)}_{1:L_r}))\Bigg)- \\ \log\Bigg(\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w}))\Bigg)\Bigg] \tag{5.17}$$

where the first, *numerator*, term is the logarithm of the unnormalised posterior and second, *denominator*, term is the logarithm of the normalisation term $Z(\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha})$. The CML objective function in equation (5.17) is a non-concave function, possibly having several local maxima [128, 169]. Alternatively, if the equivalent of Viterbi training and decoding (Section 5.2.2) is performed then the SCRF/CAug posterior is given by equation (5.13) which, when substituted into equation (5.16), yields the following form [169]

$$\mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\Bigg[\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\widehat{\mathbf{a}},\mathbf{w}^{(r)}_{1:L_r})- \\ \log\Bigg(\sum_{\{\mathbf{w},\widehat{\mathbf{a}}\}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\widehat{\mathbf{a}},\mathbf{w}))\Bigg)\Bigg] \tag{5.18}$$

where $\{\mathbf{w},\widehat{\mathbf{a}}\}$ denotes a pair consisting of word sequence $\mathbf{w}$ and associated given segmentation $\widehat{\mathbf{a}}$. Compared to equation (5.17), the CML objective function in equation (5.18) is a concave function, having a *global maximum* [73, 169].

Directly optimising either of the objective functions above with the SCRF/CAug is complicated due to the summation over word sequences [128, 281]. A related problem with the HMM was addressed by means of the *lattice framework* (Section 2.6.2). The objective functions in equations (5.17) and (5.18) may be expressed in the lattice framework as [128, 281]

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \log([[\mathbb{L}^{(r)}_{\texttt{num}}]]) - \log([[\mathbb{L}^{(r)}_{\texttt{den}}]]) \tag{5.19}$$

where $\mathbb{L}^{(r)}_{\texttt{num}}$ and $\mathbb{L}^{(r)}_{\texttt{den}}$ are the numerator and denominator lattice respectively, $[[\cdot]]$ is the lattice weight. The numerator lattice weight with the SCRF/CAug may be expressed as [128, 281]

$$[[\mathbb{L}^{(r)}_{\texttt{num}}]] = \sum_{\mathbf{a} \in \mathbb{L}^{(r)}_{\texttt{num}}} \widetilde{P}(a_1|a_0) \prod_{s=1}^{|\mathbf{a}|} \exp(\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a_s\}}, a_s)) \widetilde{P}(a_{s+1}|a_s) \tag{5.20}$$

where $\mathbf{a}$ represents a word/phone arc sequence. If the equivalent of Viterbi training and decoding is performed then the number of arc sequences in $\mathbb{L}^{(r)}_{\texttt{num}}$ is constrained to one [128]. The optimisation of this objective function may be performed using standard multivariate optimisation techniques, such as RProp and gradient ascent, [128, 281]. The gradient may be expressed as

$$\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \nabla_{\boldsymbol{\alpha}} \log([[\mathbb{L}^{(r)}_{\texttt{num}}]]) - \nabla_{\boldsymbol{\alpha}} \log([[\mathbb{L}^{(r)}_{\texttt{den}}]]) \tag{5.21}$$

It has been observed however that directly optimising this objective function may cause generalisation issues [85, 128, 281]. In order to address this issue, a prior $P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}})$ with parameters $\boldsymbol{\alpha}^{\texttt{p}}$, such as Gaussian, may be introduced [73, 281]. The final objective function may be expressed as [278]

$$\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D}) = \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D}) + \log(P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}})) \tag{5.22}$$

The gradient of $\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D})$ is a sum of $\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}; \mathcal{D})$ in equation (5.21) and $\nabla_{\boldsymbol{\alpha}}$ $\log(P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}}))$. For Gaussian priors, the corresponding expression was given in equation (3.19).

As an example consider optimising the acoustic model parameters $\boldsymbol{\alpha}_{\texttt{am}}$ associated with acoustic segment features $\boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a_s\}}, a_s)$. The gradient of $\log([[\mathbb{L}^{(r)}_{\texttt{num}}]])$

with respect to $\boldsymbol{\alpha}_{\texttt{am}}$ may be expressed as [128, 281]

$$\nabla_{\boldsymbol{\alpha}_{\texttt{am}}} \log([[\mathbb{L}^{(r)}_{\texttt{num}}]]) = \sum_{\mathbf{a}\in\mathbb{L}^{(r)}_{\texttt{num}}} P(\mathbf{a}|\mathbf{w}^{(r)}_{1:L_r}, \mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\alpha}) \sum_{s=1}^{|\mathbf{a}|} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a_s\}}, a_s) \tag{5.23}$$

where

$$P(\mathbf{a}|\mathbf{w}^{(r)}_{1:L^{(r)}}, \mathbf{O}^{(r)}_{1:T^{(r)}}; \boldsymbol{\alpha}) = \\ \frac{1}{Z(\mathbf{O}^{(r)}_{1:T_r}, \mathbf{w}^{(r)}_{1:L_r}; \boldsymbol{\alpha})} \widetilde{P}(a_1|a_0) \prod_{s=1}^{|\mathbf{a}|} \exp(\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a_s\}}, a_s)) \widetilde{P}(a_{s+1}|a_s) \tag{5.24}$$

The contribution of given arc a_s towards the gradient is given by features $\boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a_s\}}, a_s)$ weighted by the posterior associated with the given arc $\gamma_{a_s} = P(a_s|\mathbf{w}^{(r)}_{1:L_r}, \mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\alpha})$ [128]. The summation over arc sequences in equation (5.23) may then be simplified in the SCRF/CAug to the summation over individual word/phone arcs [128]

$$\nabla_{\boldsymbol{\alpha}_{\texttt{am}}} \log\left([[\mathbb{L}^{(r)}_{\texttt{num}}]]\right) = \sum_{a\in\mathbb{L}^{(r)}_{\texttt{num}}} \gamma_a \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a\}}, a) \tag{5.25}$$

The arc posterior γ_a may be computed using equation (2.67) based on the equivalents of HMM forward α_a and backward β_a arc probabilities. These forward and backward arc ”probabilities” may be computed using similar to equation (2.65) and (2.66) recursions [128, 281]

$$\alpha_a = \exp(\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a\}}, a; \boldsymbol{\lambda}))^{\kappa} \sum_{a'\text{preceding } a} \alpha_{a'} \widetilde{P}(a|a') \tag{5.26}$$

$$\beta_a = \sum_{a'\text{following } a} \exp(\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\phi}(\mathbf{O}^{(r)}_{\{a'\}}, a'; \boldsymbol{\lambda}))^{\kappa} \beta_{a'} \widetilde{P}(a'|a) \tag{5.27}$$

where κ is the acoustic de-weighting constant discussed in Section 2.7.2.1. The denominator lattice, $\mathbb{L}^{(r)}_{\texttt{den}}$, may be handled in the same way, yielding $\nabla_{\boldsymbol{\alpha}} \log([[\mathbb{L}^{(r)}_{\texttt{den}}]])$, [128, 281]. The optimisation may then be performed as discussed above.

5.3.2 Optimisation of MBR

The structured discriminative model parameter estimation based on the MBR criterion may be performed by minimising the following objective function [73]

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha})\mathcal{L}(\mathbf{w},\mathbf{w}_{1:L_r}^{(r)}) \tag{5.28}$$

where $\mathcal{L}(\cdot)$ is a loss function. The SCRF/CAug can associate loss with individual segmentations $\mathbf{a}$ giving the following variant [169]

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha})\mathcal{L}(\mathbf{a},\mathbf{w}_{1:L_r}^{(r)}) \tag{5.29}$$

Rather than minimising loss it is possible to maximise accuracy similar to the MPE estimation of HMM parameters in Section 2.7.2.2 and MBR estimation of MaxEnt parameters in Section 3.1.2.2. This yields the following variant[128, 169]

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{w},\mathbf{a}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})\mathcal{A}(\mathbf{a},\mathbf{w}_{1:L_r}^{(r)}) \tag{5.30}$$

Following [184], this objective function will be called *minimum word error* (MWE) or *minimum phone error* (MPE) depending on whether the accuracy function is computed at the word or phone level.

Optimising SCRF/CAug parameters based on the MWE/MPE objective function in equation (5.30) can be computationally expensive [128, 282]. In order to address computational issues, the lattice framework discussed in Sections 2.6.2 and 5.3.1 can be adopted [128, 282]. The MWE/MPE objective function can be expressed in the lattice framework as [128, 169]

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\frac{\sum\limits_{\mathbf{a}\in\mathbb{L}_{\texttt{den}}^{(r)}}\widetilde{P}(a_1|a_0)\prod\limits_{s=1}^{|\mathbf{a}|}\exp(\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)},a_s))\widetilde{P}(a_{s+1}|a_s)\mathcal{A}(\mathbf{a},\mathbf{w}_{1:L_r}^{(r)})}{\sum\limits_{\mathbf{a}\in\mathbb{L}_{\texttt{den}}^{(r)}}\widetilde{P}(a_1|a_0)\prod\limits_{s=1}^{|\mathbf{a}|}\exp(\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)},a_s))\widetilde{P}(a_{s+1}|a_s)} \tag{5.31}$$

where the numerator term weighs each arc sequence $\mathbf{a}$ by the associated accuracy $\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)})$, the denominator term is the denominator lattice weight $[[\mathbb{L}_{\texttt{den}}^{(r)}]]$. The optimisation can be performed using standard multivariate optimisation techniques, such as RProp and gradient ascent, [128, 282]. In order to address possible generalisation issues [128], a prior $P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}})$ with parameters $\boldsymbol{\alpha}^{\texttt{p}}$, such as Gaussian, may be introduced [73]. The final objective function can be expressed as [278]

$$\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D}) = \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) + \log(P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}})) \tag{5.32}$$

The gradient of $\mathcal{F}(\boldsymbol{\alpha}; \mathcal{D})$ is a sum of $\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D})$ and $\nabla_{\boldsymbol{\alpha}} \log(P(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{\texttt{p}}))$. For Gaussian priors, the corresponding expression was given in equation (3.19).

As an example consider optimising the acoustic model parameters $\boldsymbol{\alpha}_{\texttt{am}}$ associated with acoustic segment features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s)$. The gradient of $\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D})$ in equation (5.29) with respect to $\boldsymbol{\alpha}_{\texttt{am}}$ can be expressed as [128, 169]

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}_{\texttt{am}}} \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{a} \in \mathbb{L}_{\texttt{den}}^{(r)}} \sum_{s=1}^{|\mathbf{a}|} P(\mathbf{a}, \mathbf{w} | \mathbf{O}_{1:T^{(r)}}^{(r)}; \boldsymbol{\alpha}) \Bigg(\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L^{(r)}}^{(r)}) - \\ \sum_{\mathbf{a}' \in \mathbb{L}_{\texttt{den}}^{(r)}} P(\mathbf{a}', \mathbf{w}' | \mathbf{O}_{1:T^{(r)}}^{(r)}; \boldsymbol{\alpha}) \mathcal{A}(\mathbf{a}', \mathbf{w}_{1:L^{(r)}}^{(r)}) \Bigg) \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s)\end{aligned} \tag{5.33}$$

where $P(\mathbf{a}, \mathbf{w} | \mathbf{O}_{1:T^{(r)}}^{(r)}; \boldsymbol{\alpha})$ is the posterior of arc sequence $\mathbf{a}$ and the associated word sequence $\mathbf{w}$. The gradient above accumulates acoustic segment features for each arc a_s in $\mathbf{a}$ and weighs them by the product of $P(\mathbf{a}, \mathbf{w} | \mathbf{O}_{1:T^{(r)}}^{(r)}; \boldsymbol{\alpha})$ and the difference between $\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L^{(r)}}^{(r)})$ and the expected accuracy of word sequences in the denominator lattice $\mathbb{L}_{\texttt{den}}^{(r)}$. A similar expression was derived for optimising the MaxEnt parameters in equation (3.23). Similar to the MPE estimation of HMM parameters (Section 2.7.2.2), the accuracy function is assumed to be decomposable over individual arcs [128, 169]

$$\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) = \sum_{s=1}^{|\mathbf{a}|} \mathcal{A}(a_s) \tag{5.34}$$

where $\mathcal{A}(a_s)$ is arc a_s accuracy. The simplest option with CAug is to set it to the approximate phone arc accuracy function $\widetilde{A}(a_s)$ in equation (2.111). The contribution of arc a_s towards the gradient is given by the observation features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s)$ weighted by the product of arc posterior γ_{a_s} and the difference between the average accuracy of arc sequences passing through the current arc and the average accuracy of all arc sequences [128]. The summation over arc sequences in equation (5.33) may then be replaced by the summation over individual arcs yielding [128]

$$\nabla_{\boldsymbol{\alpha}_{\mathtt{am}}} \mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{a \in \mathbb{L}_{\mathtt{den}}^{(r)}} \gamma_a (c_a - c^{(r)}) \boldsymbol{\phi}(\mathbf{O}_{\{a\}}^{(r)}, a) \tag{5.35}$$

where c_a is the average accuracy of arc sequences passing through arc a and $c^{(r)}$ is the average accuracy of arc sequences. These accuracies may be computed based on the equivalents of HMM forward α'_a and backward β'_a correctness using equation (2.107) and (2.108) [128]. These forward and backward correctnesses may be computed using similar to equation (2.109) and (2.110) recursions [128]

$$\alpha'_a = \frac{\sum\limits_{a' \text{ preceding } a} \alpha_{a'} \widetilde{P}(a|a') \alpha'_{a'}}{\sum\limits_{a' \text{ preceding } a} \alpha_{a'} \widetilde{P}(a|a')} + \mathcal{A}(a) \tag{5.36}$$

$$\beta'_a = \frac{\sum\limits_{a' \text{ following } a} \widetilde{P}(a'|a) \exp(\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a'\}}^{(r)}, a')) \beta_{a'} (\beta'_{a'} + \mathcal{A}(a'))}{\sum\limits_{a' \text{ following } a} \widetilde{P}(a'|a) \exp(\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a'\}}^{(r)}, a'))^{\kappa} \beta_{a'}} \tag{5.37}$$

The gradient in equation (5.35) has a similar form to the weak-sense auxiliary function in equation (2.104) used for MWE/MPE estimation of HMM parameters. The optimisation may then be performed as discussed above.

5.3.3 Optimisation of large margin

The structured discriminative model parameter estimation based on the large margin criterion may be performed by maximising the following objective function

[73]

$$\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\left[\max_{\mathbf{w}\neq\mathbf{w}^{(r)}_{1:L_r}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{1:L_r}) - \log\left(\frac{P(\mathbf{w}^{(r)}_{1:L_r}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha}))}{P(\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha}))}\right)\right\}\right]_{+} \tag{5.38}$$

where $[\cdot]$ is the hinge loss function given in equation (2.77). In contrast to the CML and MBR criteria which are based on the posterior, the use of *posterior ratio* means that the issues associated with computing normalisation terms are not relevant to the large margin criterion: during training the normalisation term cancels and during inference it does not alter the rank ordering [73]. The optimisation of this objective function is closely linked to the margin-based estimation of HMM parameters in Section 2.7.1.4 and large margin estimation of MaxEnt parameters in Section 3.1.2.3 and have been examined with the structured discriminative models, such as the SCRF/CAug [277, 278].

As was discussed in Section 5.2, depending on how the segmentation is handled, the SCRF/CAug posterior may be expressed based on a single or multiple segmentations. Using single segmentation $\widehat{\mathbf{a}}$ yields a *convex optimisation problem* [73], similar to the CML estimation in Section 5.3.1. The large margin objective function based on *single segmentation* may then be expressed as [277]

$$\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\Bigg[-\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\widehat{\mathbf{a}},\mathbf{w}^{(r)}_{1:L_r})+ \max_{\{\mathbf{w}'\neq\mathbf{w}^{(r)}_{1:L_r},\widehat{\mathbf{a}}'\}}\left\{\mathcal{L}(\mathbf{w}',\mathbf{w}^{(r)}_{1:L_r}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\widehat{\mathbf{a}}',\mathbf{w}')\right\}\Bigg] \tag{5.39}$$

Comparing the large margin objective functions in equation (3.25) and (5.38), there are several differences in optimising MaxEnt and SCRF/CAug parameters. First, the segmentations must be available for each reference, $\mathbf{w}^{(r)}_{1:L^{(r)}}$, and competing, $\mathbf{w}' \neq \mathbf{w}^{(r)}_{1:L^{(r)}}$, word sequences. As was discussed in Section 5.2.2, the simplest option is to adopt segmentations provided by the generative classifier [278]; alternatively, these may be inferred [276, 277] using the semi-Markov variant of Viterbi algorithm (Section 5.2.2). Second, finding the most competing word sequence and segmentation may be computationally expensive. When the loss function is computed as a sum of segment-level losses, similar to the accuracy function in MPE estimation of HMM parameters (Section 2.7.2.2) and MBR estimation of SCRF/CAug parameters (Section 5.3.3) in equation (5.34), then an efficient Viterbi-style algorithm (Section 5.2.2) may be formulated [275, 277, 278]. Similar to other criteria discussed in this section, the final objective function incorporates a Gaussian prior, which may help to address possible generalisation issues [277, 278]. Optimising the SCRF/CAug parameters may then be performed similar to the MaxEnt (Section 3.1.2.3) using the approaches discussed in [277].

Similar to how the large margin trained MaxEnt was related in Section 3.2.3.2 to a multi-class support vector machines (SVM), the large margin trained SCRF/CAug can be related [277, 278] to a *structured SVM* [238, 267].

5.4 Adaptation to speaker and noise

In order to work reliably in real-world applications any speech recognition system must be designed to be robust to changes in speaker and noise conditions. For HMMs, maximum entropy models and SVMs related approaches were discussed in Section 2.8, 3.1.3 and 3.2.4 respectively. There has been some previous work on adapting HCRF and SCRF/CAug to speaker and noise conditions.

The use of maximum-a-posteriori (MAP) adaptation, discussed with HMMs in Section 2.8 and MaxEnt in Section 3.1.3, has been also investigated with HCRF [222]. Being a general adaptation scheme it makes no assumption about the nature of the feature-functions. However, when these exhibit some structure the use of other adaptation schemes may be more advantageous [73].

Alternatively, the use of linear transformation based adaptation has been investigated with HCRF in [221]. This scheme makes use of approaches similar to the maximum likelihood linear regression (MLLR) for HMMs discussed in Section 2.8.1 and linear feature transform for MaxEnt discussed in Section 3.1.3. As these schemes have been applied only with feature-functions resembling those used in HMMs, it is not clear whether this form of adaptation approaches can be extended to more general feature-functions [73].

Finally, the feature-functions can be modified to make them dependent on the speaker and noise conditions [73]. The structured discriminative models then can be trained speaker and noise independent similar to the standard, unstructured discriminative models in Chapter 3. This approach will be discussed in Chapter 6.

5.5 Summary

This chapter has discussed structured discriminative models. These models introduce a structure into the discriminative classifier to address situations where the number of possible sentences is large. A similar approach was earlier discussed with hidden Markov models (HMM) in Section 2.3.2. There were two primary structures discussed. The first form follows HMMs in relating observations with hidden states. Examples given included maximum entropy Markov models (MEMM) and hidden conditional random fields (HCRF). The second form relates segments with words/phones. Examples given included segmental conditional random fields (SCRF) and conditional augmented models (CAug). A range of aspects have been examined with these models including handling latent variables relating observations and segments with states and words/phones respectively, parameter estimation and adaptation.

Chapter 6

Feature-functions

The previous chapters have assumed the existence of appropriate feature-functions. The selection of these feature-functions is known to be central to the performance of all discriminative models examined so far in this thesis [128, 168, 217, 281]. The feature-functions can be broadly split into frame-level, acoustic segment and supra-segmental feature-functions [73]. The *frame-level feature-functions* discussed next in Section 6.1 act on the current frame or a fixed span of frames surrounding the current frame to extract features. The MEMM/HCRF features in equation (5.3) are simple examples of frame-level features. In contrast to frame-level feature-functions, the *acoustic segment feature-functions* discussed in Section 6.2 act on all the observations associated with a segment. The MaxEnt features in equation (3.12), which are based on the MEMM/HCRF features summed over all the observations associated with the segment, are simple examples of acoustic segment features. Finally, the *supra-segmental feature-functions* discussed in Section 6.3 act on the state, phone or word sequences. The SCRF/CAug pronunciation and language model log-probabilities in equation (5.9) are simple examples of supra-segmental features.

6.1 Frame-level features

The simplest form of feature-functions are restricted to those extracting features at the frame level [73]. One example of frame-level features is given by [73]

$$\boldsymbol{\phi}(\mathbf{o}_t, a_s) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i)\mathbf{o}_t \\ \delta(a_s^{\mathtt{i}}, v_i)\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}} \\ \vdots \end{bmatrix} \tag{6.1}$$

where a_s is a segment with label $a_s^{\mathtt{i}}$ spanning the observation vector $\mathbf{o}_t$, v_i is a segment identity and $\mathcal{V}$ is a vocabulary of segment identities. These features are the same as the MEMM/HCRF features in equation (5.3) if each segment a_s refers to a hidden state q_t and the same as the MaxEnt features in equation (3.12) if there is only one segment spanning the entire sequence of observations. It is possible to show [90, 91] that using these features, which bear a resemblance to the HMM mean and covariance statistics in equations (2.49) and (2.50) (see Section 2.2.3), yields structured discriminative models related to discriminatively trained HMMs discussed in Section 2.7 [73].

A number of variations on this basic form can be considered. One variation extends the features with higher-order statistics [256]. Another variation splices the current static observation and a fixed window of the previous and future static observations together, possibly transforming [93], to form the observation vector, rather than using the complete observation vector in equation (2.6) containing the current frame and optionally one or more dynamic parts [73].

It is also possible to apply classifiers to the observation vector to provide bottom-up information on where the frames lie in a pseudo-linguistic space [73]. This may come in the form of indicator features

$$\boldsymbol{\phi}(\mathbf{o}_t, a_s) = \begin{bmatrix} \vdots \\ \delta(u_t, v_i) \\ \vdots \end{bmatrix} \tag{6.2}$$

where u_t is a linguistic unit at time t, and/or class posterior features

$$\boldsymbol{\phi}(\mathbf{o}_t, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(u_t, v_i)P(v_i|\mathbf{o}_t; \boldsymbol{\lambda}) \\ \vdots \end{bmatrix} \tag{6.3}$$

where $\boldsymbol{\lambda}$ are the classifier parameters, which may then be either directly appended to or used in place of the features in equation (6.1) [73]. Examples of classifiers examined include multilayer perceptrons providing the class posterior probability of phone units [163], Gaussians of an HMM-based recogniser providing the (sparse) class posterior probabilities of HMM state-components [93, 257] and HMM-based recognisers on the complete observation sequence providing indication on the linguistic unit the current frame belongs to [281].

6.2 Acoustic segment features

The frame-level feature-functions discussed in Section 6.1 will generate T feature vectors for a T-length observation sequence. An alternative option is to devise feature-functions acting on all the observations associated with a segment. This served the basis of segmental conditional random fields (SCRF) and conditional augmented (CAug) models in Section 5.1. The fundamental requirement imposed on the acoustic segment feature-functions is that they must map variable-length observation sequences into fixed-length feature-space [73]. The simplest example of acoustic segment feature-function was given in equation (5.8) where a frame-level feature-function was summed over time indices associated with the segment

$$\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s) = \sum_{t \in \{a_s\}} \boldsymbol{\phi}(\mathbf{o}_t, a_s) \tag{6.4}$$

Other examples include *score-spaces* [101, 216] and, similar in spirit [73], *event detectors* [281, 283]. This rest of this section will focus on the score-spaces.

6.2.1 Score-spaces

The score-space may be defined as a model-based feature-space [216]. Many score-spaces are based on generative models [73, 217]. Given a generative model, there are options to define the score-space [145, 216]. One option is to make use of generative model parameters re-estimated on the available segment of observations [145, 146]. For HMMs discussed in Chapter 2, the use of means, covariances and (C)MLLR transform parameters have been considered [27, 53, 194, 219]. An alternative option and the one examined in this section is to make use of functions associated with these models such as log-likelihood and derivatives of log-likelihood with respect to generative model parameters [101, 216].

The rest of this section is organised as follows. The following Section 6.2.1.1 gives several examples of score-spaces making use of log-likelihoods and derivatives. The next Section 6.2.1.2 discusses dependencies possible to incorporate into the discriminative model using these score-spaces. The last Section 6.2.1.3 discusses how the discriminative models based on these score-spaces can be trained speaker and noise independent.

6.2.1.1 Examples

A wide range of score-spaces making use of log-likelihoods and derivatives of log-likelihood with respect to generative model parameters have been proposed in the literature [101, 128, 216]. These differ in the form of generative models and the way these functions are combined to form score-space.

For instance, one of them, a *Fisher score-space* [101], is based on the global, class-independent generative model and makes use of derivatives with respect to generative model parameters $\boldsymbol{\lambda}$ to define the score-space

$$\boldsymbol{\phi}_{\mathtt{f}}(\mathbf{O}_{\{a_s\}};\boldsymbol{\lambda}) = \Big[\nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}};\boldsymbol{\lambda}))\Big] \tag{6.5}$$

Another example is a *likelihood ratio score-space* [216]

$$\boldsymbol{\phi}_{\mathtt{r}}(\mathbf{O}_{\{a_s\}};\boldsymbol{\lambda}) = \begin{bmatrix} \log\left(\dfrac{p(\mathbf{O}_{\{a_s\}}|\omega_1;\boldsymbol{\lambda})}{p(\mathbf{O}_{\{a_s\}}|\omega_2;\boldsymbol{\lambda})}\right) \\ \nabla_{\boldsymbol{\lambda}}\log(p(\mathbf{O}_{\{a_s\}}|\omega_1;\boldsymbol{\lambda})) \\ -\nabla_{\boldsymbol{\lambda}}\log(p(\mathbf{O}_{\{a_s\}}|\omega_2;\boldsymbol{\lambda})) \end{bmatrix} \tag{6.6}$$

which extends on the Fisher score-space by introducing the class-specific generative model, and the log-likelihood ratio into the score-space. The Fisher and likelihood ratio score-space in the form of dynamic kernel in equation (3.45) have been examined with the SVM (Section 3.2.2) in [101, 128, 216].

The use of derivative features may not always be possible [278]. One option to address this issue is to construct the score-space based only on log-likelihoods. An example of such score-space is an *appended likelihood score-space* [216]

$$\boldsymbol{\phi}_{\mathtt{a}}(\mathbf{O}_{\{a_s\}}, a_s;\boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i)\begin{bmatrix} \log(p(\mathbf{O}_{\{a_s\}}|v_1;\boldsymbol{\lambda})) \\ \vdots \\ \log(p(\mathbf{O}_{\{a_s\}}|v_{|\mathcal{V}|};\boldsymbol{\lambda})) \end{bmatrix} \\ \vdots \end{bmatrix} \tag{6.7}$$

For each class v_i, the appended likelihood score-space incorporates the log-likelihood given the current class, v_i, as well as the log-likelihoods given all competing classes, $v \neq v_i$. The appended score-space have been examined with the multi-class SVM and SCRF/CAug in [278]. One interesting aspect of this score-space is that the multi-class SVM and SCRF/CAug become closely related to the HMM [278]. For instance, inferring the class maximising the MaxEnt posterior can be expressed based on equations (3.10) and (6.7) as

$$\widehat{a}_s = \arg\max_{v_i}\left\{\sum_{j=1}^{|\mathcal{V}|}\alpha^{(v_i,v_j)}\log(p(\mathbf{O}_{\{a_s\}}|v_j;\boldsymbol{\lambda}))\right\} \tag{6.8}$$

where $\alpha^{(v_i,v_j)}$ is the class v_i parameter associated with the log-likelihood of $\mathbf{O}_{\{a_s\}}$ given class v_j. If $\alpha^{(v_i,v_j)}$ is set to one when $v_i = v_j$ and zero otherwise then the class maximising the MaxEnt posterior becomes the class maximising the HMM likelihood. In practice this relationship offers the multi-class SVM and SCRF/CAug opportunity for initialising parameters in the way which guarantees the HMM classification performance [278]. Note that this may not be possible with other forms of feature-functions.

As the number of classes increases the use of appended score-spaces may become impractical [275]. One option to address this issue is to retain for each class v_i only those features which are derived based on the corresponding generative model. This gives rise to a *likelihood score-space* [216]

$$\boldsymbol{\phi}_{\mathtt{l}}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s, v_i) \log(p(\mathbf{O}_{\{a_s\}}|v_i; \boldsymbol{\lambda})) \\ \vdots \end{bmatrix} \tag{6.9}$$

Compared to the appended likelihood score-space, the likelihood score-space contain $|\mathcal{V}|$ times less features. A variant of this score-space, which incorporates the log-likelihood from more than one generative model, has been examined with the SCRF/CAug in [94]. In addition to the log-likelihood, it is possible to add derivatives which give rise to the *first-order likelihood score-space* [216]

$$\boldsymbol{\phi}_{\mathtt{l}}^{(1)}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s, v_i) \log(p(\mathbf{O}_{\{a_s\}}|v_i; \boldsymbol{\lambda})) \\ \delta(a_s, v_i) \nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}}|v_i; \boldsymbol{\lambda})) \\ \vdots \end{bmatrix} \tag{6.10}$$

The first-order likelihood score-space has been examined with the SCRF/CAug in [128]. Both these score-spaces when applied with the multi-class SVM or SCRF/CAug can guarantee the HMM classification performance. Higher than the first-order likelihood score-spaces can be defined analogously [216].

The use of log-likelihoods, first- and higher-order derivatives quickly increases the number of features. Table 6.1 provides dimensionality for a number of score-spaces considered in this section. The first score-space examined was the Fisher score-space $\boldsymbol{\phi}_{\mathtt{f}}$ in equation (6.5) whose dimensionality equals to the number of pa-

Score-space	Dimensionality
$\boldsymbol{\phi}_{\mathtt{f}}$	$\dim(\boldsymbol{\lambda})$
$\boldsymbol{\phi}_{\mathtt{r}}$	$\dim(\boldsymbol{\lambda}) + 1$
$\boldsymbol{\phi}_{\mathtt{a}}$	$\lvert\mathcal{V}\rvert^2$
$\boldsymbol{\phi}_{\mathtt{l}}$	$\lvert\mathcal{V}\rvert$
$\boldsymbol{\phi}_{\mathtt{l}}^{(1)}$	$\dim(\boldsymbol{\lambda}) + \lvert\mathcal{V}\rvert$
$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$	$\dim(\{\boldsymbol{\mu}_{j,m}\}) + \lvert\mathcal{V}\rvert$

Table 6.1: Dimensionality of selected score-spaces based on generative models with parameters $\boldsymbol{\lambda}$ and $|\mathcal{V}|$ classes. (The discriminative and generative models are assumed to share the same definition of classes. The generative model parameters are assumed not to be tied across classes)

rameters in the generative model, $\dim(\boldsymbol{\lambda})$. The log-likelihood ratio score-space $\boldsymbol{\phi}_{\mathtt{r}}$ in equation (6.6) is similar to the Fisher score-space yet one additional dimension, the log-likelihood ratio, is used. The appended likelihood score-space $\boldsymbol{\phi}_{\mathtt{a}}$ in equation (6.7) has dimensionality equal to the square of the vocabulary size, $|\mathcal{V}|^2$, as the dimensionality of feature sub-spaces for each vocabulary element v equals to the size of vocabulary $|\mathcal{V}|$. The likelihood score-space $\boldsymbol{\phi}_{\mathtt{l}}$ in equation (6.9) makes use of a single feature, log-likelihood, for each element of vocabulary. Thus, the dimensionality of the likelihood score-space is equal to the size of vocabulary. The first-order likelihood score-space $\boldsymbol{\phi}_{\mathtt{l}}^{(1)}$ in equation (6.10), which combines the aspects of likelihood and Fisher score-spaces, has dimensionality equal to the size of vocabulary plus the number of generative model parameters.

Empirically it has been observed that generalisation improves when features believed to be the most discriminative are only selected [229]. The derivatives with respect to HMM mean vectors are often cited as the most discriminative first-order derivatives [128, 217]. For example, the first-order likelihood score-space, $\boldsymbol{\phi}_{\mathtt{l}}^{(1)}$, based only on these derivatives, also known as the *HMM mean derivative* *score-space*, has the following form

$$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s, v_i)\log(p(\mathbf{O}_{\{a_s\}}|v_i; \boldsymbol{\lambda})) \\ \vdots \\ \delta(a_s, v_i)\nabla_{\boldsymbol{\mu}_{j,m}}\log(p(\mathbf{O}_{\{a_s\}}|v_i; \boldsymbol{\lambda})) \\ \vdots \end{bmatrix} \tag{6.11}$$

The HMM mean derivative score-space in contrast to the first-order likelihood score-space incorporates derivatives with respect to mean vector rather than all parameters. The dimensionality of this score-space thus is equal to the size of vocabulary plus the size of the set of mean vector parameters $\dim(\{\boldsymbol{\mu}_{j,m}\})$. Note that this and other numbers given in this section are based on assumptions that generative and discriminative models share the same definition of classes and that generative model parameters are not tied. As will be discussed in Chapter 7, a difference in the definition of generative and discriminative model classes (e.g. states in generative model and phones in discriminative model) and the use of tying may cause various issues such as the tree intersect effect (see Section 2.4) which may impair generalisation in these models. The HMM mean derivative $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$, likelihood $\boldsymbol{\phi}_{\mathtt{l}}$, appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ and likelihood ratio $\boldsymbol{\phi}_{\mathtt{r}}$ score-spaces will be examined in Chapter 8.

6.2.1.2 Dependencies

The choice of generative model is fundamental to the score-spaces [73] as it determines independence assumptions that can not be overcome and conditional independence assumptions that can be overcome [128].

When the HMMs with parameters $\boldsymbol{\lambda} = \{\ldots, c_{j,m}, \boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}, \ldots\}$ are used as the generative model these conditional independence assumptions are the state and observation conditional independence assumptions discussed in Section 2.2. The derivatives of the HMM log-likelihood with respect to the component weight

$c_{j,m}$, mean $\boldsymbol{\mu}_{j,m}$ and covariance $\boldsymbol{\Sigma}_{j,m}$ are given by [128, 145]

$$\nabla_{c_{j,m}} \log(p(\mathbf{O}_{\{a_s\}}|v_i;\boldsymbol{\lambda}) = \sum_{t\in\{a_s\}} \frac{\gamma_{j,m}(t)}{c_{j,m}} - \gamma_j(t) \tag{6.12}$$

$$\nabla_{\boldsymbol{\mu}_{j,m}} \log(p(\mathbf{O}_{\{a_s\}}|v_i;\boldsymbol{\lambda}) = \sum_{t\in\{a_s\}} \gamma_{j,m}(t)\boldsymbol{\Sigma}^{-1}_{j,m}(\mathbf{o}_t - \boldsymbol{\mu}_{j,m}) \tag{6.13}$$

$$\nabla_{\boldsymbol{\Sigma}_{j,m}} \log(p(\mathbf{O}_{\{a_s\}}|v_i;\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t\in\{a_s\}} \gamma_{j,m}(t)(-\boldsymbol{\Sigma}^{-1}_{j,m} + \boldsymbol{\Sigma}^{-1}_{j,m}(\mathbf{o}_t - \boldsymbol{\mu}_{j,m})(\mathbf{o}_t - \boldsymbol{\mu}_{j,m})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}_{j,m}) \tag{6.14}$$

where $\gamma_j(t) = P(q^j_t|\mathbf{O}_{\{a_s\}}, v_i; \boldsymbol{\lambda})$ and $\gamma_{j,m}(t) = P(q^{j,m}_t|\mathbf{O}_{\{a_s\}}, v_i; \boldsymbol{\lambda})$ are the state and state-component occupancies defined in Section 2.2 by equations (2.39) and (2.40). These occupancies are functions of the complete observation sub-sequence $\mathbf{O}_{\{a_s\}}$ which ”breaks” the HMM conditional independence assumptions [128].

Alternatively, the use of GMMs with parameters $\boldsymbol{\lambda} = \{\ldots, c_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \ldots\}$ as the generative model yields frame-level features [73]. For instance, the derivative of the GMM log-likelihood with respect to component weight c_m is given by [145]

$$\nabla_{c_j} \log(p(\mathbf{O}_{\{a_s\}}|v_i;\boldsymbol{\lambda})) = \sum_{t\in\{a_s\}} \frac{P(q^j_t|\mathbf{o}_t, v_i;\boldsymbol{\lambda})}{c_j} - 1 \tag{6.15}$$

Apart from the use of scaling and shift, these features are the same [73] as the class posterior features [93, 257] in equation (6.3).

Higher than the first-order derivatives may introduce even more complex features [128]. For instance, the second-order derivative of the HMM log-likelihood with respect to component weights $c_{j,m}$ and $c_{k,n}$ is given by [64, 128]

$$\nabla_{c_{k,n}}\nabla_{c_{j,m}} \log(p(\mathbf{O}_{\{a_s\}}|v_i;\boldsymbol{\lambda}) = -\frac{2}{c_{j,m}c_{k,n}} \sum_{t\in\{a_s\}} \delta(j,k)\delta(m,n)\gamma_{j,m}(t) + \tag{6.16}$$

$$\sum_{t\in\{a_s\}}\sum_{\tau\in\{a_s\}} \frac{D(q^{j,m}_t, q^{k,n}_\tau) - c_{j,m}D(q^j_t, q^{j,m}_t) - c_{k,m}D(q^{j,m}_t, q^k_\tau) + c_{j,m}c_{k,n}D(q^j_t, q^k_\tau)}{c_{j,m}c_{k,n}}$$

where

$$D(q^{j,m}_t, q^{k,n}_\tau) = P(q^{j,m}_t, q^{k,n}_\tau|\mathbf{O}_{\{a_s\}}, v_i;\boldsymbol{\lambda}) - \gamma_{j,m}(t)\gamma_{k,n}(\tau) \tag{6.17}$$

and $P(q_t^{j,m}, q_\tau^{k,n}|\mathbf{O}_{\{a_s\}}, v_i; \boldsymbol{\lambda})$ is a posterior probability of state-component pair $\{q_t^{j,m}, q_\tau^{k,n}\}$. These posteriors in case when $t \neq \tau$ and/or $j \neq k$ allow explicit dependencies between the two discontiguous in time and/or space components to be modelled [128].

In order to illustrate the potential usefulness of score-spaces consider a simple two-class (1 and -1) two-symbol (`A` and `B`) problem, where the training data $\mathcal{D}$ = $\{\{1, \texttt{AAAA}\}, \{1, \texttt{BBBB}\}, \{-1, \texttt{AABB}\}, \{-1, \texttt{BBAA}\}\}$ and the generative models are discrete HMMs with topology shown in Figure 6.1 [128]. The maximum likelihood (ML) estimation discussed in Section 2.2.3 for both classes yields identical HMM parameter estimates also shown in Figure 6.1. As shown by the first, $\log p$, row in Table 6.2, the log-likelihood features derived from these HMMs can not correctly classify the training data [128]. On the other hand, the use of the first- and second-order derivative features may help to discriminate between classes by taking advantage of additional dependencies [128]. As shown by the second, $\nabla_{2,\texttt{A}}$, and third, $\nabla_{2,\texttt{A}}\nabla_{2,\texttt{A}}$, row in Table 6.2, the first- and second-order derivatives of log-likelihood with respect to symbol `A` in state 2 make the training data separable though non-linearly [64]. As shown by the last, $\nabla_{2,\texttt{A}}\nabla_{3,\texttt{A}}$, row in Table 6.2, the

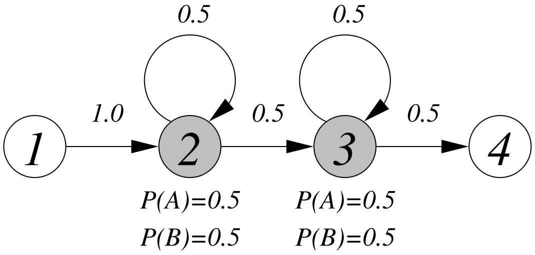


Figure 6.1: Example discrete HMM topology

Features	Class 1		Class -1	
	AAAA	BBBB	AABB	BBAA
$\log p$	-4.44	-4.44	-4.44	-4.44
$\nabla_{2,\texttt{A}}$	0.50	-0.50	0.33	-0.33
$\nabla_{2,\texttt{A}}\nabla_{2,\texttt{A}}$	-3.83	0.17	-3.28	-0.61
$\nabla_{2,\texttt{A}}\nabla_{3,\texttt{A}}$	-0.17	-0.17	-0.06	-0.06

Table 6.2: Example log-likelihood and selected derivative features

use of second-order derivative with respect to symbol A in state 2 and 3 makes the training data linearly separable [128]. The latter derivative captures the obvious difference between the two classes that the symbol changes part way through [64].

6.2.1.3 Adaptation and compensation framework

As was discussed in Sections 3.1.3, 3.2.4 and 5.4, the alternative approach to adapting the discriminative classifiers to speaker and noise conditions is to modify the feature-functions. When the score-spaces are based on generative models this can be achieved using model-based adaptation/compensation schemes [68]. When the HMM is used as the generative model then the examples of model-based adaptation/compensation schemes include (constrained) maximum likelihood linear regression (MLLR) discussed in Section 2.8.1 and vector Taylor series (VTS) discussed in Section 2.8.2.

The general score-space adaptation/compensation framework [68] is illustrated by Figure 6.2. The shaded part in Figure 6.2 shows the model-based adapta-

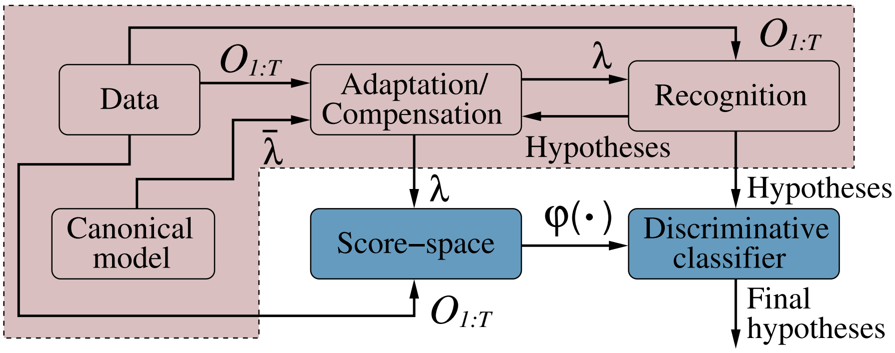


Figure 6.2: Adaptation/compensation scheme for discriminative classifiers using score-spaces based on generative models

tion/compensation stage. Given observation sequence $\mathbf{O}_{1:T}$, the canonical model parameters $\bar{\boldsymbol{\lambda}}$ are modified to match the target speaker and noise conditions yielding the adapted model with parameters $\boldsymbol{\lambda}$. A score-space in the unshaded part of Figure 6.2 makes use of the adapted model to yield modified feature vectors for the discriminative classifier. The discriminative classifiers examined in this framework include the SVM [68], multi-class SVM [278] and SCRF/CAug [278].

6.3 Supra-segmental features

The last type of features examined in this chapter are supra-segmental features which are primarily associated with state, phone or word sequences [73] and may provide with various sorts of information such as lexical [5, 197, 281], syntactic [5, 36] and semantic [6, 32, 121].

The use of supra-segmental features in this thesis has so far been focused on the transition/pronunciation $\boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L})$ and language model $\boldsymbol{\phi}(\mathbf{w}_{1:L})$ features in equations (5.4) and (5.9). One common form of these features is based on the bag-of-word model [110] and higher-order n-grams. For instance, the unigram and bigram features may be expressed as [5, 134, 197, 254, 281]

$$\boldsymbol{\phi}(a^{\mathtt{i}}_{s-1}, a^{\mathtt{i}}_{s}) = \begin{bmatrix} \vdots \\ \delta(a^{\mathtt{i}}_{s}, v_i) \\ \delta(a^{\mathtt{i}}_{s-1}, v_h)\delta(a^{\mathtt{i}}_{s}, v_i) \\ \vdots \end{bmatrix} \tag{6.18}$$

where the segment labels $a^{\mathtt{i}}_{s-1}$ and $a^{\mathtt{i}}_{s}$ may correspond to states, phones or words. The unigram and bigram features can be adopted to provide, for instance, the language model features

$$\boldsymbol{\phi}(\mathbf{w}_{1:L}) = \sum_{l=1}^{L} \boldsymbol{\phi}(w_{l-1}, w_l) \tag{6.19}$$

It is also possible to apply various linguistic and statistical tools to the labels to provide bottom-up information on where the labels lie in a pseudo-linguistic space, similar in spirit to the frame-level feature-functions discussed in Section 6.1. One common form is based on indicator functions

$$\boldsymbol{\phi}(a^{\mathtt{i}}_{s-1}, a^{\mathtt{i}}_{s}) = \begin{bmatrix} \vdots \\ \delta(u_{s}, v_i) \\ \delta(u_{s-1}, v_h)\delta(u_{s}, v_i) \\ \vdots \end{bmatrix} \tag{6.20}$$

where u_s is a linguistic unit associated with the segment a_s. Examples of tools examined include parsers providing part-of-speech labels [5, 29, 36], morphological analysers providing lemma, root and stem-ending labels [5, 210], topic classifiers providing topic sensitive labels [6, 121] and clustering algorithms providing word class category labels [32].

6.4 Summary

This chapter has provided an overview of feature-functions that have been examined with the standard, unstructured, and structured discriminative classifiers discussed in Chapters 3 and 5 respectively. These feature-functions were split into frame-level, acoustic segment and supra-segmental feature-functions.

The frame-level feature-functions is the simplest form of feature-functions extracting features at the frame level. Examples of the frame-level features included the first- and second-order observation statistics similar to the HMM mean and covariance statistics discussed in Chapter 2.

In contrast, the acoustic segment feature-functions act on a variable-length sequence of observations associated with a segment. These feature-functions must satisfy the fundamental requirement of mapping variable-length sequences to fixed length first mentioned in Chapter 3. Examples of the acoustic segment features included a number of score-spaces based on generative models, such as the HMM.

Finally, the supra-segmental feature-functions primarily act on sequences of words, phones and/or states to provide lexical, syntactic and semantic information. Examples of supra-segmental features given included unigram and bigram features which provide part-of-speech, lemma, root and stem-ending, topic and word class statistics.

Chapter 7

Conditional augmented models

Chapter 5 discussed several structured discriminative models proposed for speech recognition tasks. These models aim to incorporate standard approach in speech recognition of partitioning sentence into words, and words into phones into a discriminative model framework [73]. Some of these models, such as segmental conditional random fields, adopt structuring into word units and have been applied to a range of tasks [281, 283]. Others, such as conditional augmented models, aim to adopt a deeper structuring including phone units but have so far been applied to small vocabulary tasks based on monophone [128] or word [278] units. In order to make conditional augmented models models more generally applicable, the two directions need to be combined. The combined model could adopt context-dependent phone-level acoustic and word-level language and pronunciation modelling. Issues that need to be addressed with this model include handling of context-dependent phones and robust parameter estimation. This chapter proposes a number of approaches to address these issues.

7.1 Overview

The work presented in this chapter adopts the following form of posterior probability of word sequence $\mathbf{w}_{1:L}$ given observation sequence $\mathbf{O}_{1:T}$

$$P(\mathbf{w}_{1:L}|\mathbf{O}_{1:T};\boldsymbol{\alpha},\boldsymbol{\lambda}) = \frac{1}{Z(\mathbf{O}_{1:T};\boldsymbol{\alpha},\boldsymbol{\lambda})}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T},\mathbf{a},\mathbf{w}_{1:L};\boldsymbol{\lambda})) \tag{7.1}$$

where summation is performed over all possible segmentations into word and phone sequences, the discriminative model parameters $\boldsymbol{\alpha}$ and features $\boldsymbol{\phi}(\cdot)$ are given by

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_{\texttt{am}} \\ \boldsymbol{\alpha}_{\texttt{pm}} \\ \boldsymbol{\alpha}_{\texttt{lm}} \end{bmatrix}, \quad \boldsymbol{\phi}(\mathbf{O}_{1:T}, \mathbf{a}, \mathbf{w}_{1:L}) = \begin{bmatrix} \sum_{s=1}^{|\mathbf{a}|} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) \\ \boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L}) \\ \boldsymbol{\phi}(\mathbf{w}_{1:L}) \end{bmatrix} \tag{7.2}$$

The suprasegmental features, $\boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L})$ and $\boldsymbol{\phi}(\mathbf{w}_{1:L})$, (Section 6.3) provide the pronunciation and language model features. These are often set to the standard pronunciation and n-gram language model probabilities

$$\boldsymbol{\phi}(\mathbf{a}, \mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{a}^{\mathtt{i}}|\mathbf{w}_{1:L}))\Big], \quad \boldsymbol{\phi}(\mathbf{w}_{1:L}) = \Big[\log(P(\mathbf{w}_{1:L}))\Big] \tag{7.3}$$

similar to the example given with the SCRF/CAug in Section 5.1. The pronunciation $\alpha_{\texttt{pm}}$ and language $\alpha_{\texttt{lm}}$ model parameters are set to the standard scaling factors. Although it is possible to incorporate a range of acoustic segment features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda})$, the work presented in this chapter will focus on the score-spaces based on generative models with parameters $\boldsymbol{\lambda}$, such as the HMM discussed in Chapter 2. As discussed in Section 6.2, these score-spaces can provide a rich set of features capable of introducing long-term dependencies, can be adapted to speaker and noise conditions, which yields speaker and noise independent discriminative models (Section 6.2.1.3), simply by adapting generative model using model-based approach (Section 2.8). Thus, there are two sets of parameters in CAug: the discriminative model $\boldsymbol{\alpha}$ and generative model $\boldsymbol{\lambda}$ parameters.

Apart from the inclusion of generative model parameters, the adopted form of posterior is essentially the same as the form discussed with the SCRF/CAug in Section 5.1. The previous work with the form of model in equation (7.1) have considered associating discriminative model parameters with individual words or monophones (context-independent phones). A range of parameter estimation criteria have been considered, such as conditional maximum likelihood (CML) and variants of minimum Bayes' risk (MBR) and large margin discussed in Section 5.3. The equivalents of HMM Viterbi and forward-backward algorithms have also been proposed as discussed in Section 5.2. In order to adapt to particular speaker and noise conditions schemes have also been proposed as discussed in Section 5.4.

This chapter extends the previous work with the SCRF/CAug to incorporate context-dependent phone classes into the structured discriminative model. The following Section 7.2 discusses extensions needed to define score-spaces based on the context-dependent generative models. The next Section 7.3 discusses how the discriminative model parameters associated with the context-dependent phone classes can be tied to improve the robustness of estimates obtained using the CML, MBR or large margin criteria. The next to last Section 7.4 discusses how the generative model parameters can be re-estimated using the standard discriminative and discriminative adaptive criteria. The last Section 7.5 provides a summary of the whole chapter.

7.2 Context-dependent score-spaces

The previous work with CAug has considered the use of monophones [128] or words [278]. When context-dependent phones are considered then the dimensionality of some score-spaces discussed in Section 6.2.1 may become very large. This section discusses the scalability of some of those score-spaces and gives illustrative examples of how many discriminative model parameters may be required with the context-dependent phones.

As discussed in Section 6.2.1, the appended likelihood score-space $\boldsymbol{\phi}_{\mathtt{a}}$ scales quadratically with the number of context-dependent phones, $|\mathcal{V}|$. For the context-dependent phone $a_s^{\mathtt{i}}$ associated with segment a_s and comprising observation subsequence $\mathbf{O}_{\{a_s\}}$, the appended likelihood score-space would require computing log-likelihood given *all* context-dependent phone classes $v_1, \ldots, v_{|\mathcal{V}|}$ as shown by

$$\boldsymbol{\phi}_{\mathtt{a}}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i) \begin{bmatrix} \log(p(\mathbf{O}_{\{a_s\}}|v_1; \boldsymbol{\lambda})) \\ \vdots \\ \log(p(\mathbf{O}_{\{a_s\}}|v_{|\mathcal{V}|}; \boldsymbol{\lambda})) \end{bmatrix} \\ \vdots \end{bmatrix} \tag{7.4}$$

For the triphone set based on 40 monophones, $|\mathcal{V}| = 40^3$, this would require estimating approximately 4 billion parameters.

Alternative score-spaces may require fewer parameters. For instance, the likelihood score-space $\boldsymbol{\phi}_{\mathtt{1}}$ excludes log-likelihoods given all but the *current* context-dependent phone $a_s^{\mathtt{i}}$ as shown below

$$\boldsymbol{\phi}_{\mathtt{1}}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i) \log(p(\mathbf{O}_{\{a_s\}}|a_s^{\mathtt{i}}; \boldsymbol{\lambda})) \\ \vdots \end{bmatrix} \tag{7.5}$$

The use of this score-space in the above example would require estimating 64 thousand parameters. Rather than excluding log-likelihoods given all competing context-dependent phones as in equation (7.5) it is possible to exclude all but few context-dependent phones. One issue with this approach is that it is not obvious which context-dependent phones may provide log-likelihoods useful for discrimination. In this work a simple idea is examined where context-dependent phones examined for each context-dependent phone are those which share or *match* the same context. An example below shows context-dependent phones that will be examined for triphone `sil-dh+iy`[1]

$$\begin{bmatrix} \texttt{sil-aa+iy} \\ \vdots \\ \textbf{sil-dh+iy} \\ \vdots \\ \texttt{sil-z+iy} \end{bmatrix} \tag{7.6}$$

Note that all these triphones differ in the central phone yet share or match the same left and right context. The total number of triphones in equation (7.6) equals to the number of monophones. When the current context-dependent phone changes then so does the set of examined context-dependent phones to match the new context. For example, the following sets of triphones will be examined for

[1] Phones such as silence and short pause may also be incorporated. For these phones no context-dependent phones should be used.

the triphone sequence `sil-dh+iy`, `dh-iy+d`, `iy-d+ao`, ..., `s-t+sil`[1]

$$\begin{bmatrix} \texttt{sil-aa+iy} \\ \vdots \\ \textbf{sil-dh+iy} \\ \vdots \\ \texttt{sil-z+iy} \end{bmatrix}, \begin{bmatrix} \texttt{dh-aa+d} \\ \vdots \\ \textbf{dh-iy+d} \\ \vdots \\ \texttt{dh-z+d} \end{bmatrix}, \begin{bmatrix} \texttt{iy-aa+ao} \\ \vdots \\ \textbf{iy-d+ao} \\ \vdots \\ \texttt{iy-z+ao} \end{bmatrix}, \dots, \begin{bmatrix} \texttt{s-aa+sil} \\ \vdots \\ \textbf{s-t+sil} \\ \vdots \\ \texttt{s-z+sil} \end{bmatrix} \tag{7.7}$$

The score-space illustrated above in the following will be called the *matched-context score-space* $\boldsymbol{\phi}_{\mathtt{m}}$. The use of this score-space in the above example would require estimating approximately 3 million parameters.

The use of higher than zero order score-spaces increases dimensionality of score-spaces further by incorporating the first and higher order derivatives of the log-likelihood given each context-dependent phone. For example, the use of first-order likelihood score-space as shown below

$$\boldsymbol{\phi}_{\mathtt{l}}^{(1)}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i) \begin{bmatrix} \log(p(\mathbf{O}_{\{a_s\}}|a_s^{\mathtt{i}}; \boldsymbol{\lambda})) \\ \nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}}|a_s^{\mathtt{i}}; \boldsymbol{\lambda})) \end{bmatrix} \\ \vdots \end{bmatrix} \tag{7.8}$$

would require $\dim(\boldsymbol{\lambda}^{(a_s^{\mathtt{i}})}) + 1$ parameters for the current context-dependent phone $a_s^{\mathtt{i}}$, where $\boldsymbol{\lambda}^{(a_s^{\mathtt{i}})}$ are the generative model parameters associated with $a_s^{\mathtt{i}}$. If the generative model parameters are not tied then the total number of discriminative acoustic model parameters to estimate equals $\dim(\boldsymbol{\lambda}) + |\mathcal{V}|$. For an HMM with N emitting states, M components per state and d-dimensional Gaussian state-component output densities with diagonal covariance matrices, the number of HMM parameters is given by $(N+2)^2 + NM(1+2d)$ where the first term gives the number of parameters associated with transition probabilities and the second term gives the number of parameters associated with state output density parameters. Thus, a total of $((N+2)^2 + NM(1+2d))|\mathcal{V}| + |\mathcal{V}|$ discriminative

[1] This triphone sequence represents one possible expansion of sentence `the dog chased the cat` discussed in Section 2.3.1 into triphone HMM units.

acoustic model parameters would be required. For the set of triphone HMMs where $N = 3$, $M = 12$, $d = 39$ and $|\mathcal{V}| = 40^3$ (assuming 40 monophones) this would yield approximately 200 million parameters.

The above examples show that the number of discriminative acoustic model parameters can become large when the CAug model features are based on the context-dependent phones. Furthermore, as discussed in Section 2.3.1, the use of context-dependent units makes it hard to obtain good coverage in the training data. For instance, the use of cross-word units, such as the triphones discussed in this section, typically yields a large number of units with few if any examples [266]. In order to address the data sparsity problem, an approach is required to ensure that there is sufficient training data to robustly estimate the discriminative acoustic model parameters.

7.3 Parameter tying

For small vocabulary tasks, where whole-word generative models are used, the discriminative model parameters may be associated with the individual words. This is the approach adopted with CAug in the previous work [128, 278]. For larger vocabulary tasks, where state-level phonetic decision tree tying is often used to determine context-dependent generative models, the appropriate tying of the discriminative model parameters is less clear as the discriminative acoustic model introduces conditional independence assumptions at the model rather than state level. If there is sufficient training data then the discriminative model parameters could be specified at the context-dependent phone level, as determined by the state-level decision trees. However, it is not possible to guarantee that all context-dependent phones are observed in the training data, as the tying operates not at the model but state level.

In order to address this problem, a *model-level parameter tying* is performed to determine the appropriate tying of the discriminative model parameters. The approach based on the model-level phonetic decision tree tying (Section 2.4) is used. However, a special care is required as the generative model parameters are themselves tied at the state-level. When using two distinct decision trees, it is possible to get a *tree-intersect* style approach discussed in Section 2.4, where the

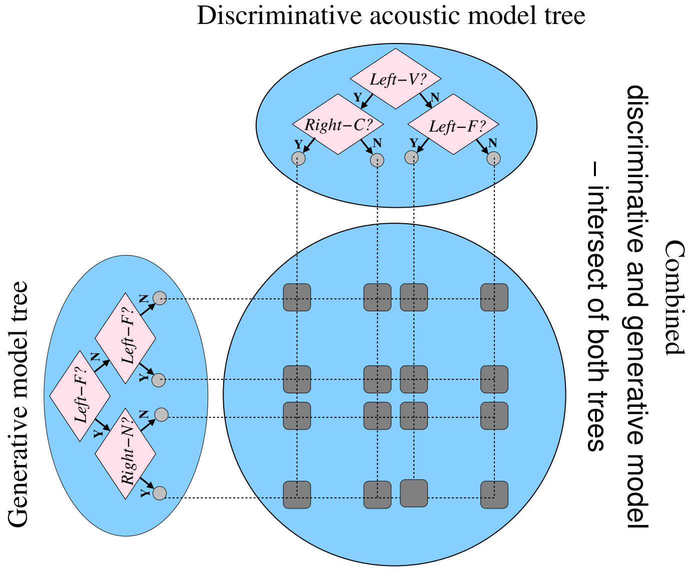

Discriminative acoustic model tree
Left−V?
Right−C?
Left−F?
Y
N
Generative model tree
Left−F?
Left−F?
Right−N?
Combined
discriminative and generative model
– intersect of both trees

Figure 7.1: A tree-intersect model based on discriminative and generative model trees. The individual physical, discriminative and generative, models are shown by shaded circles. The combined physical models are shown at the intersect of two trees by shaded squares

effective number of parameters becomes very large. This may result in robustness issues when training the combined model. Figure 7.1 shows an example where discriminative and generative model trees contain 4 leaf nodes. The combined model is formed by intersecting the two trees which gives a total of 16 possible parameter values, twice the number of parameters available to those trees (8).

In order to address this problem, the decision tree for the discriminative acoustic model is built based on those context-dependent phones that appear at the leaf nodes of the decision trees created for the generative models. The leaves of this tree can be guaranteed to have a minimum occupancy count in the training data and at least one distinct state. A consequence of this approach is that the maximum number of context-dependent phone classes for the discriminative acoustic model is the number of distinct generative models.

Figure 7.1 may be used to illustrate how the combined model parameters are determined for any context-dependent phone in two steps. The first step drops the context-dependent phone down the decision trees on the left to yield the generative model parameters. The second step drops the label associated with the generative model parameters down the decision tree at the top to yield the discriminative acoustic model parameters.

The procedure outlined above is sub-optimal in many respects. The first issue is that the labels associated with the generative model parameters may correspond to multiple context-dependent phones of which one (physical) is chosen to represent the rest (logical). In this work the most frequently occurring context-dependent phone was used as the label. The second issue is that the clustering procedure is insensitive to the choice of score-spaces. A consequence of this is that the order of features is not taken into account. The order of features plays an important role in, for example, first-order derivative and higher score-spaces. The generative models, such as HMMs, do not typically maintain consistent order of components. Figure 7.2 shows two states where the same three components are arranged in a different way. Although log-likelihoods computed with these models are not affected[1], the situation is different with derivatives as these depend

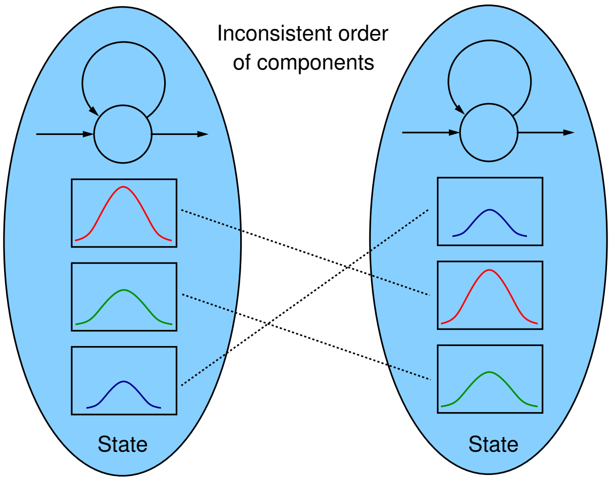


Figure 7.2: An example of inconsistent order of HMM components. Shown are two single state HMMs sharing 3 components ordered in different ways (dotted arrows connect identical components). The log-likelihood (single-dimensional feature) does not depend on the order of components and hence will be the same for both HMMs. The derivatives (three-dimensional features) do depend on the order of components and hence will be different up to permutation of components.

[1] Log-likelihoods computed with these models will be identical as the order of Gaussian components in mixtures does not affect the result. Hence these models yield identical log-likelihood features.

on the order of components and hence yield different features. This may result in "strong" features being masked by other, "weaker" features. The greater the level of tying applied the worse this masking may be. In order to address this problem, the features can be summed within the states as shown below for the HMM mean derivative score-space in equation (6.11)

$$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\overline{\mu})}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) = \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i) \begin{bmatrix} \vdots \\ \sum\limits_{m \in S_j} \nabla_{\boldsymbol{\mu}_{j,m}} \log(p(\mathbf{O}_{\{a_s\}} | a_s^{\mathtt{i}}; \boldsymbol{\lambda})) \\ \vdots \end{bmatrix} \\ \vdots \end{bmatrix} \tag{7.9}$$

In addition to reducing the number of parameters and possibly improving the robustness of estimates, this approach can be also helpful in dealing with situations where the number of components is not consistent across the HMM states.

7.4 Generative model parameter estimation

The joint estimation of discriminative and generative model parameters is complicated [128]. Instead, a sequential optimisation has been adopted where given fixed generative model parameters, the discriminative model parameters are estimated [128, 278]. Section 5.3 discussed how the discriminative model parameters can be estimated using a range of discriminative criteria, such as conditional maximum likelihood (CML) and minimum word/phone error (MWE/MPE) variants of minimum Bayes' risk (MBR) criterion. The previous discussion, however, has not considered how the generative model parameters can be re-estimated given the discriminative model parameters.

The previous chapter showed that score-spaces make use of generative models in different ways. For instance, the likelihood $\boldsymbol{\phi}_{\mathtt{l}}$ and appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ score-spaces (see below) make use of log-likelihoods computed with these models. On the other hand, the first-order likelihood score-space in equation (6.10), in addition to the log-likelihood, makes use of derivatives with respect to generative model parameters. In addition, there are a range of generative models to choose from, such as the GMM, the HMM and the trajectory HMM. These generative models may employ constrained parameters, such as HMM transition probabilities, which are constrained to sum to one for each state, and HMM covariance matrices, which must be positive semi-definite. The parameters may also come in a non-canonical form, being adapted to speaker and noise conditions using model-based adaptation/compensation schemes, such as the maximum likelihood linear regression (MLLR) and vector Taylor series (VTS) discussed in Section 2.8. Thus, if discriminative adaptive training is to be used, the underlying canonical model parameters must be considered.

Chapter 2 discussed that *extended Baum-Welch* (EBW) update rules can be derived with the HMM in the standard discriminative (Sections 2.7.2.1 and 2.7.2.2) and adaptive training (Sections 2.8.1.2 and 2.8.2.4) scenarios. One advantage of these update rules is that the HMM parameter constraints will be automatically satisfied. In order to derive these update rules for the MMI and MPE estimation of HMM parameters, an approach based on *weak-sense auxiliary functions* was discussed in Section 2.7.2. These, in contrast to strong-sense auxiliary functions, such as the one used for ML estimation of HMM parameters in Section 2.2.3, do not guarantee convergence of the underlying objective function. However, if the weak-sense auxiliary function converges then it does so to the local maximum of the objective function [184].

This section shows that similar to EBW update rules can be also derived for the CML and MWE/MPE estimation of HMM parameters with CAug which adopts the likelihood score-space $\boldsymbol{\phi}_{\mathtt{l}}$ in equation (7.5) and the appended likelihood score-space $\boldsymbol{\phi}_{\mathtt{a}}$ in equation (7.4). Discriminative and discriminative adaptive training of generative model parameters with these forms of CAug are closely linked with the corresponding HMM approaches in Sections 2.7.1 and 2.8. The rest of this section is organised as follows. The CML estimation of generative model parameters is discussed in details in Section 7.4.1. The following Section 7.4.2 builds upon the previous section to show how MWE/MPE estimation of HMM parameters can be performed with CAug. The last Section 7.4.3 discusses modification required for CML and MPE estimation of HMM parameters in the CMLLR-based discriminative speaker adaptive training (CMLLR-DSAT) and discriminative VTS adaptive training (DVAT).

7.4.1 Optimisation based on CML

The objective function to maximise for CML estimation of generative model parameters $\boldsymbol{\lambda}$ with CAug can be expressed as

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log(P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})) \tag{7.10}$$

where $\mathcal{D}$ is the supervised training data, $\boldsymbol{\alpha}$ are discriminative model parameters. Similar to the MMI estimation of HMM parameters (Section 2.7.2.1) and CML estimation of MaxEnt (Section 3.1.2.1) and SCRF/CAug (Section 5.3.1) parameters, the objective function can be re-written as the difference of two terms

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) - \mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) \tag{7.11}$$

where

$$\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}; \boldsymbol{\lambda}))\right) \tag{7.12}$$

is the numerator term and

$$\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))\right) \tag{7.13}$$

is the denominator term. Given the CML objective function, consider a weak-sense auxiliary function for the numerator term $\mathcal{G}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$, where $\boldsymbol{\lambda}$ and $\widehat{\boldsymbol{\lambda}}$ are the current and new HMM parameters respectively. This weak-sense auxiliary function can then be combined with the weak-sense auxiliary function for the denominator term $\mathcal{G}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ to yield the weak sense auxiliary function for the CML objective function $\mathcal{G}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$.

The weak-sense auxiliary function for the numerator term must satisfy

$$\nabla_{\widehat{\boldsymbol{\lambda}}}\mathcal{G}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})\Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} = \nabla_{\widehat{\boldsymbol{\lambda}}}\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})\Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} \tag{7.14}$$

The gradient of $\mathcal{F}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ is given by (Section A.1)

$$\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \nabla_{\boldsymbol{\lambda}} \{\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} \tag{7.15}$$

where $\boldsymbol{\alpha}_{\mathtt{am}}$ are discriminative acoustic model parameters. One suitable form for the weak-sense auxiliary function

$$\mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s, a_s^{\mathtt{i}}}^{(r)}) \tag{7.16}$$

where $\mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)} = \{\{\mathbf{O}_{\{a_s\}}^{(r)}, a_s^{\mathtt{i}}\}\}$ is the supervised training data consisting of observation sub-sequence $\mathbf{O}_{\{a_s\}}^{(r)}$ and identity $a_s^{\mathtt{i}}$, and $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)})$ is an auxiliary function which satisfies

$$\nabla_{\widehat{\boldsymbol{\lambda}}} \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)}) \Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} = \nabla_{\widehat{\boldsymbol{\lambda}}} \{\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \widehat{\boldsymbol{\lambda}})\} \Big|_{\widehat{\boldsymbol{\lambda}}=\boldsymbol{\lambda}} \tag{7.17}$$

This ensures that $\mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ is the weak-sense auxiliary function for $\mathcal{F}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$. The simplest option to define the auxiliary function $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)})$ with the likelihood and appended likelihood score-space is to make use of the strong-sense auxiliary function in equation (2.47) which is both strong- and weak-sense auxiliary function for the HMM log-likelihood [184]. The following form of equation (2.47) can be used to optimise mean and covariance parameters [265]

$$\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)}) = \sum_{t \in \{a_s\}} \sum_{\{j,m\}} \gamma_{a_s^{\mathtt{i}},j,m}(t) \log(\mathcal{N}(\mathbf{o}_t^{(r)}; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m})) + K \tag{7.18}$$

where K is constant in mean and covariance parameters. This yields the following form of auxiliary function for the likelihood

$$\mathcal{Q}_1(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)}) = \boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \begin{bmatrix} \vdots \\ \delta(a_s^{\mathtt{i}}, v_i) \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}_{a_s,a_s^{\mathtt{i}}}^{(r)}) \\ \vdots \end{bmatrix} \tag{7.19}$$

and the appended likelihood score-spaces

$$\mathcal{Q}_{\mathtt{a}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}^{(r)}_{a_s, a^{\mathtt{i}}_s}) = \boldsymbol{\alpha}^{\mathsf{T}}_{\mathtt{am}} \begin{bmatrix} \vdots \\ \delta(a^{\mathtt{i}}_s, v_i) \begin{bmatrix} \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a_s, v_1}) \\ \vdots \\ \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a_s, v_{|\mathcal{V}|}}) \end{bmatrix} \\ \vdots \end{bmatrix} \tag{7.20}$$

A similar derivation leads to the following form of weak-sense auxiliary function for the denominator term $\mathcal{F}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$

$$\mathcal{Q}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{w}} \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w} | \mathbf{O}^{(r)}_{1:T_r}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}^{(r)}_{a_s, a^{\mathtt{i}}_s}) \tag{7.21}$$

Subtracting $\mathcal{Q}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ from $\mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ yields the weak-sense auxiliary functions for the CML objective function $\mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$.

$$\mathcal{G}_{\mathtt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) - \mathcal{Q}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) \tag{7.22}$$

In order to address possible generalisation issues, the final form of CML objective function often incorporates a prior, such as the I-smoothing prior in equation (2.94)

$$\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \mathcal{F}_{\mathtt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, ; \mathcal{D}) + \log(p(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{\mathtt{p}})) \tag{7.23}$$

where $\boldsymbol{\lambda}^{\mathtt{p}}$ are prior parameters. The weak-sense auxiliary function for $\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ can be defined by adding the logarithm of prior on the new parameters to $\mathcal{G}_{\mathtt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$. In addition, the smoothing function $\mathcal{Q}_{\mathtt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}})$ in equation (2.87), which has zero gradient when evaluated at the current parameters $\boldsymbol{\lambda}$, can be added to $\mathcal{G}_{\mathtt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ to improve convergence as in the standard MMI/MPE estimation of HMM parameters in Sections 2.7.2.1 and 2.7.2.2. The weak-sense auxiliary function for the final objective function $\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ is given by

$$\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) - \mathcal{Q}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) + \mathcal{Q}_{\mathtt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}) + \log(p(\widehat{\boldsymbol{\lambda}}; \boldsymbol{\lambda}^{\mathtt{p}})) \tag{7.24}$$

The weak-sense auxiliary function for the MMI estimation of HMM parameters in equation (2.93) has the same form. Taking derivative of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ with respect to new mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and solving yields the following update rule

$$\widehat{\boldsymbol{\mu}}_{j,m} = \frac{\left\{\boldsymbol{\theta}^{\mathtt{num}}_{j,m} - \boldsymbol{\theta}^{\mathtt{den}}_{j,m}\right\} + D_{j,m}\boldsymbol{\mu}_{j,m} + \tau^{\mathtt{I}}\widehat{\boldsymbol{\mu}}^{\mathtt{p}}_{j,m}}{\left\{\gamma^{\mathtt{num}}_{j,m} - \gamma^{\mathtt{den}}_{j,m}\right\} + D_{j,m} + \tau^{\mathtt{I}}} \tag{7.25}$$

where $\gamma^{\mathtt{num}}_{j,m}$, $\boldsymbol{\theta}^{\mathtt{num}}_{j,m}$ and $\gamma^{\mathtt{den}}_{j,m}$, $\boldsymbol{\theta}^{\mathtt{den}}_{j,m}$ are the occupancy and mean statistics associated with the numerator and denominator term weak sense auxiliary functions, $\tau^{\mathtt{I}}$ is the I-smoothing constant. Taking derivative of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ with respect to the new covariance $\widehat{\boldsymbol{\Sigma}}_{j,m}$ and solving yields the following update rule

$$\widehat{\boldsymbol{\Sigma}}_{j,m} = \frac{\left\{\boldsymbol{\Theta}^{\mathtt{num}}_{j,m} - \boldsymbol{\Theta}^{\mathtt{den}}_{j,m}\right\} + D_{j,m}(\boldsymbol{\Sigma}_{j,m} + \boldsymbol{\mu}_{j,m}\boldsymbol{\mu}^{\mathsf{T}}_{j,m}) + \tau^{\mathtt{I}}(\widehat{\boldsymbol{\Sigma}}^{\mathtt{p}}_{j,m} + \widehat{\boldsymbol{\mu}}^{\mathtt{p}}_{j,m}\widehat{\boldsymbol{\mu}}^{\mathtt{p}^{\mathsf{T}}}_{j,m})}{\left\{\gamma^{\mathtt{num}}_{j,m} - \gamma^{\mathtt{den}}_{j,m}\right\} + D_{j,m} + \tau^{\mathtt{I}}} - \widehat{\boldsymbol{\mu}}_{j,m}\widehat{\boldsymbol{\mu}}^{\mathsf{T}}_{j,m} \tag{7.26}$$

where $\boldsymbol{\Theta}^{\mathtt{num}}_{j,m}$ and $\boldsymbol{\Theta}^{\mathtt{den}}_{j,m}$ are the covariance statistics associated with the numerator and denominator term weak sense auxiliary functions. The derivation of these update rules for the likelihood $\boldsymbol{\phi}_{\mathtt{l}}$ and appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ score-spaces follow the derivations in [184] where the strong-sense auxiliary function in equation (2.47) is replaced by the weighted strong-sense auxiliary function ($\boldsymbol{\phi}_{\mathtt{l}}$) in equation (7.19) and the weighted sum of strong-sense auxiliary functions ($\boldsymbol{\phi}_{\mathtt{a}}$) in equation (7.20) respectively. The update rules in equation (7.25) and (7.26) have the same form as the EBW update rules applied in MMI/MPE estimation of HMM parameters in equation (2.97) and (2.98). Thus, the same framework can be adopted to update the HMM parameters with these forms of CAug model.

Similar to the MMI/MPE estimation of HMM parameters, the lattice framework discussed in Sections 2.6.2 and 2.7.2 with the HMM and also in Sections 5.3.1 and 5.3.2 with the SCRF/CAug can be adopted to handle summation over segmentations and segmentations and word sequences in the numerator $\mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ and denominator $\mathcal{Q}_{\mathtt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ term weak-sense auxiliary functions respectively using numerator and denominator lattices. Noting that the contribution of each segment a_s in $\mathcal{Q}_{\mathtt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ to the gradient of the final weak-sense auxiliary function $\mathcal{G}(\boldsymbol{\alpha},\ \boldsymbol{\lambda},\ \widehat{\boldsymbol{\lambda}};\ \mathcal{D})$ is given by the gradient of auxiliary function associated with a_s, $\mathcal{Q}(\boldsymbol{\lambda},\ \widehat{\boldsymbol{\lambda}};\ \boldsymbol{\alpha},\ \mathcal{D}^{(r)}_{a_s,a^{\mathtt{i}}_s})$, weighted by the posterior probability associated with given segmentation, the summation over segmentations can be simplified to the summation over individual numerator lattice arcs

$$\mathcal{Q}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{num}}}\gamma_a\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}^{(r)}_{a,a^{\texttt{i}}}) \tag{7.27}$$

where $\mathbb{L}^{(r)}_{\texttt{num}}$ is the numerator lattice associated with the r-th observation sequence. The denominator term weak-sense auxiliary function $\mathcal{Q}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ can be handled in a similar way yielding

$$\mathcal{Q}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}}\gamma_a\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}^{(r)}_{a,a^{\texttt{i}}}) \tag{7.28}$$

Again, taking derivative of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ with respect to the new mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and the new covariance $\widehat{\boldsymbol{\Sigma}}_{j,m}$ and solving the resulting set of equations yields the following form of denominator statistics

$$\gamma^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}}\gamma_a\sum_{t\in\{a\}}\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}}\boldsymbol{\gamma}(a, q^{j,m}_t) \tag{7.29}$$

$$\boldsymbol{\theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}}\gamma_a\sum_{t\in\{a\}}\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}}\boldsymbol{\gamma}(a, q^{j,m}_t)\mathbf{o}^{(r)}_t \tag{7.30}$$

$$\boldsymbol{\Theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}}\gamma_a\sum_{t\in\{a\}}\boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}}\boldsymbol{\gamma}(a, q^{j,m}_t)\mathbf{o}^{(r)}_t\mathbf{o}^{(r)^{\mathsf{T}}}_t \tag{7.31}$$

For the likelihood score-space $\boldsymbol{\gamma}(a, q^{j,m}_t)$ is given by

$$\boldsymbol{\gamma}_{\texttt{l}}(a, q^{j,m}_t) = \begin{bmatrix} \vdots \\ \delta(a^{\texttt{i}}, v_i)\gamma_{a^{\texttt{i}},j,m}(t) \\ \vdots \end{bmatrix} \tag{7.32}$$

For each arc a this simply returns the state-component occupancy $\gamma_{a,j,m}(t)$ associated with $q^{j,m}_t$. The dot-product of $\boldsymbol{\gamma}(a, q^{j,m}_t)$ with the acoustic model parameters $\boldsymbol{\alpha}_{\mathtt{am}}$ yields weighted state-component occupancy $\alpha_{\mathtt{am}}^{(a^{\mathtt{i}})}\gamma_{a^{\mathtt{i}},j,m}(t)$. Compared to the MMI estimation of HMM parameters, the contribution of each arc a to the statistics in the likelihood score-space is additionally scaled by the corresponding discriminative acoustic model parameter $\alpha_{\mathtt{am}}^{(a^{\mathtt{i}})}$. A different form is obtained with the appended likelihood score-space

$$\boldsymbol{\gamma}_{\mathtt{a}}(a, q_t^{j,m}) = \begin{bmatrix} \vdots \\ \delta(a^{\mathtt{i}}, v_i) \begin{bmatrix} \gamma_{v_1,j,m}(t) \\ \vdots \\ \gamma_{v_{|\mathcal{V}|},j,m}(t) \end{bmatrix} \\ \vdots \end{bmatrix} \tag{7.33}$$

Here, a vector of state-component occupancies is returned where each occupancy is computed based on arc a where identity $a^{\mathtt{i}}$ is set to one of generative model classes. Even if $q_t^{j,m}$ do not belong to the HMM specified by $a^{\mathtt{i}}$ its occupancy will be featured in $\boldsymbol{\gamma}(a, q_t^{j,m})$ as occupancies based on all generative models are taken into account. Compared to the likelihood score-space, the contribution of each arc to the statistics is a weighted combination of statistics associated with all generative models rather than one.

7.4.2 Optimisation based on MWE/MPE

The objective function to minimise for minimum Bayes' risk estimation of generative model parameters $\boldsymbol{\lambda}$ with CAug can be expressed as [73]

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{L}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)}) \tag{7.34}$$

where $\mathcal{L}(\cdot)$ is the loss function (Section 2.7.1.3). This section will consider the minimum word/phone error (MWE/MPE) variant discussed in Section 5.3.2

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{w}, \mathbf{a}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) \tag{7.35}$$

where $\mathcal{A}(\cdot)$ is the accuracy function given by equation (5.34). The gradient of $\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ is given by (Section A.1.2)

$$\nabla_{\boldsymbol{\lambda}}\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}}\sum_{s=1}^{|\mathbf{a}|} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\Big(\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) - \sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)})\Big)\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} \tag{7.36}$$

Similar to the previous section, a weak-sense auxiliary function for $\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ can be defined by

$$\mathcal{G}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}}\sum_{s=1}^{|\mathbf{a}|} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\Big(\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) - \sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)})\Big)\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s, a_s^{\mathtt{i}}}^{(r)}) \tag{7.37}$$

where $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s}^{(r)})$ is the auxiliary function given by equation (7.19) and (7.20) for the likelihood and appended likelihood score-space respectively. required to satisfy the constraint in equation (7.17). The final form of objective

$$\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, ; \mathcal{D}) + \log(p(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{\mathtt{p}})) \tag{7.38}$$

and weak-sense auxiliary function

$$\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \mathcal{G}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) + \mathcal{Q}_{\texttt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}) + \log(p(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{\mathtt{p}})) \tag{7.39}$$

is constructed similar to the previous section by adding the I-smoothing prior with parameters $\widehat{\boldsymbol{\lambda}}$ given by equation (2.94) and smoothing function $\mathcal{Q}_{\texttt{sm}}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}})$ given by equation (2.87). Taking derivative of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ with respect to new mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and new covariance $\widehat{\boldsymbol{\Sigma}}$ and solving with respect to the new parameters yields update rules in the EBW form given by equations (7.25) and (7.26).

Similar to the previous section, in order to address computational issues accumulating the required statistics, the lattice framework can be adopted. Noting that the contribution of each segment a_s to the gradient of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ is given by the gradient of $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}_{a_s, a_s^{\mathtt{i}}}^{(r)})$ associated with that segment weighted by the posterior probability associated with the underlying word/phone sequence and the difference between the accuracy of the underlying sequence and the average accuracy of all sequences, the summation over arc sequences can be simplified to summation over individual arcs

$$\mathcal{G}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \gamma_a^{\texttt{mpe}}\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \boldsymbol{\alpha}, \mathcal{D}^{(r)}_{a,a^{\texttt{i}}}) \tag{7.40}$$

where $\gamma_a^{\texttt{mpe}} = \gamma_a(c_a - c^{(r)})$ is the equivalent of MPE differential (Section 2.7.2.2) discussed with the SCRF/CAug in Section 5.3.2. Apart from the use of different auxiliary function on each arc, the weak-sense auxiliary function in the MPE estimation of HMM parameters given by equation (2.104) has the same form. Taking derivative of $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D})$ with respect to new mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and new covariance $\widehat{\boldsymbol{\Sigma}}_{j,m}$ and solving the resulting set of equations yields the following denominator statistics

$$\gamma^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma_a^{\texttt{mpe}}) \sum_{t\in\{a\}} \boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\gamma}(a, q_t^{j,m}) \tag{7.41}$$

$$\boldsymbol{\theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma_a^{\texttt{mpe}}) \sum_{t\in\{a\}} \boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\gamma}(a, q_t^{j,m})\mathbf{o}_t^{(r)} \tag{7.42}$$

$$\boldsymbol{\Theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R}\sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma_a^{\texttt{mpe}} \sum_{t\in\{a\}} \boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\gamma}(a, q_t^{j,m})\mathbf{o}_t^{(r)}\mathbf{o}_t^{(r)^{\mathsf{T}}} \tag{7.43}$$

where occupancies $\boldsymbol{\gamma}(a, q_t^{j,m})$ collected by the likelihood and appended likelihood score-space are given by equation (7.32) and (7.33) respectively. Compared to the MPE estimation of HMM parameters, the contribution of each arc a to the statistics is the statistics associated with the underlying generative model weighted by the corresponding discriminative acoustic model parameter $\alpha_{\texttt{am}}^{(a^{\texttt{i}})}$, whereas in the appended likelihood score-space this is a weighted combination of the statistics associated with all generative models.

7.4.3 Optimisation of CMLLR-based DSAT and DVAT

When the HMM parameters come in a non-canonical form, being adapted to speaker and noise conditions using model-based adaptation or compensation schemes, such as the constrained maximum likelihood linear regression (CMLLR) and vector Taylor series (VTS) discussed in Section 2.8, then the update rules derived above for the CML and MWE/MPE estimation of HMM parameters can not be used. Instead, a new set of update rules must be derived.

Similar to the previous two sections, the optimisation of CMLLR-transformed or VTS-compensated HMM parameters with CAug is closely linked with the discriminative CMLLR-based speaker adaptive training (DSAT) in Section 2.8.1.2 and discriminative VTS adaptive training (DVAT) in Section 2.8.2.4. The derivations in this section directly follow the respective derivations in the CMLLR-DSAT [253], DVAT [55, 65, 66] and the previous two sections.

For the CMLLR-based DSAT with CAug, the CMLLR-based SAT auxiliary function in equation (2.130) is adopted for the CML and MWE/MPE estimation of HMM parameters

$$\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a,a^{\mathtt{i}}}) = \sum_{t\in\{a\}} \sum_{\{j,m\}} \gamma_{a^{\mathtt{i}},j,m}(t) \log(\mathcal{N}(\overline{\mathbf{o}}^{(r)}_t; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m})) + K \tag{7.44}$$

where $\gamma_{a^{\mathtt{i}},j,m}(t)$ is the state-component posterior obtained with $\boldsymbol{\lambda}$, $\overline{\mathbf{o}}^{(r)}_t$ is the transformed observation vector based on the training data observation vector $\mathbf{o}^{(r)}_t$ and speaker transform parameters as illustrated by equation (2.128), K is constant with respect to mean and covariance parameters. Replacing $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a,a^{\mathtt{i}}})$ in equations (7.19) and (7.20) by the form given in equation (7.44), taking derivative of the weak-sense auxiliary function for the MWE/MPE objective function in equation (7.39) and solving with respect to the new canonical mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and covariance $\widehat{\boldsymbol{\Sigma}}_{j,m}$ yields update equations in the EBW form given by equations (7.25) and (7.26), where the denominator statistics is given by

$$\gamma^{\mathtt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\mathtt{den}}} \max(0, -\gamma^{\mathtt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\mathtt{am}} \boldsymbol{\gamma}(a, q^{j,m}_t) \tag{7.45}$$

$$\boldsymbol{\theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\gamma}(a, q^{j,m}_{t}) \bar{\mathbf{o}}^{(r)}_{t} \tag{7.46}$$

$$\boldsymbol{\Theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\gamma}(a, q^{j,m}_{t}) \bar{\mathbf{o}}^{(r)}_{t} \bar{\mathbf{o}}^{(r)\mathsf{T}}_{t} \tag{7.47}$$

Compared to the previous section, the statistics above is based on the transformed observation vectors, similar to the CMLLR-based MPE SAT estimation of HMM parameters in Section 2.8.1.2 yet its form is different as discussed in Section 7.4.2. Conducting a similar derivation for the CMLLR-based CML SAT estimation of HMM parameters with CAug yields update rules in the same EBW form, where the statistics in Section 7.4.1 is now based on the transformed observation vectors.

For the DVAT with CAug, the VAT auxiliary function in equation (2.180) is adopted for the CML and MWE/MPE estimation of HMM parameters

$$\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a,a^{\texttt{i}}}) = \sum_{t\in a} \sum_{\{j,m\}} \gamma_{a^{\texttt{i}},j,m}(t) \mathcal{E}\{\log(\mathcal{N}(\bar{\mathbf{o}}^{(r)}_{t}; \widehat{\boldsymbol{\mu}}_{j,m}, \widehat{\boldsymbol{\Sigma}}_{j,m})) | \mathbf{o}^{(r)}_{t}, q^{j,m}_{t}\} + K \tag{7.48}$$

Replacing $\mathcal{Q}(\boldsymbol{\lambda}, \widehat{\boldsymbol{\lambda}}; \mathcal{D}^{(r)}_{a,a^{\texttt{i}}})$ in equations (7.19) and (7.20) by the form given in equation (7.48), taking derivative of the weak-sense auxiliary function for the MWE/MPE objective function in equation (7.39) and solving with respect to the new canonical mean $\widehat{\boldsymbol{\mu}}_{j,m}$ and covariance $\widehat{\boldsymbol{\Sigma}}_{j,m}$ yields update equations in the EBW form, where the denominator statistics is given by

$$\gamma^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\gamma}(a, q^{j,m}_{t}) \tag{7.49}$$

$$\boldsymbol{\theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\gamma}(a, q^{j,m}_{t}) \mathcal{E}\{\bar{\mathbf{o}}^{(r)}_{t} | \mathbf{o}^{(r)}_{t}, q^{j,m}_{t}\} \tag{7.50}$$

$$\boldsymbol{\Theta}^{\texttt{den}}_{j,m} = \sum_{r=1}^{R} \sum_{a\in\mathbb{L}^{(r)}_{\texttt{den}}} \max(0, -\gamma^{\texttt{mpe}}_{a}) \sum_{t\in\{a\}} \boldsymbol{\alpha}^{\mathsf{T}}_{\texttt{am}} \boldsymbol{\gamma}(a, q^{j,m}_{t}) \mathcal{E}\{\bar{\mathbf{o}}^{(r)}_{t} \bar{\mathbf{o}}^{(r)\mathsf{T}}_{t} | \mathbf{o}^{(r)}_{t}, q^{j,m}_{t}\} \tag{7.51}$$

Compared to the two previous sections, the statistics is based on the expectations of transformed observation vectors given training observation vectors in equations (2.184) and (2.185), similar to the DVAT estimation of HMM parameters in Section 2.8.2.4 yet its form is different as discussed in Section 7.4.2. Conducting a similar derivation for the CML VAT estimation of HMM parameters with CAug yields update rules in the EBW form, where the statistics in Section 7.4.1 is now based on the expectations of transformed observation vectors given training observation vectors.

7.5 Summary

This chapter has considered the use of context-dependent phone units in conditional augmented (CAug) models. These structured discriminative models usually employ score-spaces based on generative models to provide features. However, the dimensionality of these score-spaces increases significantly with the use of context-dependent generative models. A new form of score-space was proposed, which provides a balance between those score-spaces that provide few and those which provide too many features. In order to address robustness issues when estimating parameters of many context-dependent phone classes from the limited amount of training data, the use of parameter tying was proposed, similar to the standard practice with generative models such as HMMs. The use of score-space based on generative models, in addition to the possibility of training speaker- and noise-independent discriminative model parameters using the score-space adaptation/compensation framework, opens an opportunity to re-estimate the underlying generative model parameters to yield more informative score-spaces. This chapter derived update rules in the extended Baum-Welch (EBW) form by means of weak-sense auxiliary functions for conditional maximum likelihood (CML) and minimum word/phone error (MWE/MPE) estimation of HMM parameters given the likelihood and appended score-space. The use of adaptively trained generative models, such as (discriminative) CMLLR-based SAT and (discriminative) VTS adaptively trained HMMs, was also considered and similar EBW update rules were derived.

Chapter 8

Experimental results

In this chapter, experimental results using extended acoustic code-breaking (Chapter 4) and conditional augmented models (Chapter 7) are presented. In order to illustrate the properties of these two approaches, two tasks were examined. The first task, Toshiba in-car data, was used with extended acoustic code-breaking. This task provides opportunity to examine standard, unstructured, discriminative models trained on artificially generated data when limited or no real examples of words such as city names exist. The second task, Aurora 4, was used with conditional augmented models. This medium-to-large vocabulary task provides opportunity to examine phone-level structured discriminative models, score-space based on context-dependent generative models and the use of discriminative model parameter tying. For a faster development time, a small vocabulary task, Aurora 2, was used to narrow down the range of options available with these two approaches, such as synthesis approaches with extended acoustic code-breaking or parameter estimation criteria with conditional augmented models.

8.1 Experimental setup

This section provides a short description of Toshiba in-car, Aurora 4 and 2 tasks. It also details software used and extended to permit experiments with extended acoustic code-breaking and conditional augmented models.

8.1.1 Toshiba in-car

Toshiba in-car [140] is a small-to-medium noise-corrupted speech recognition task. The training set is available in two conditions: clean and multi-style. The clean training data is the speaker-independent part of Wall Street Journal data [182] consisting of two subsets, WSJ0 and WSJ1, of which 284 speakers uttering approximately 66 hours of speech were used in total (WSJ SI284). The multi-style training data was obtained by artificially corrupting the clean training data with the in-car noise recordings of SpeechDat [160] and Toshiba [140] yielding the average signal-to-noise ratio (SNR) of 23 dB. The test data consisted of digit string and city name test sets. The digit string test set contains 824, 862 and 898 phone numbers spoken by 15 male and 15 female US-English speakers whilst in vehicles with engine idle (IDLE), driving in cities (CITY) and on the highway (HWAY). The average SNR is 35, 25 and 18 dB respectively. The city name subset contains 928 and 988 city names, out of 550 city names possible in total, spoken by 15 male and 15 female US-English speakers whilst in vehicles with engine idle (IDLE) and on the highway (HWAY). The average SNR is 35 and 18 dB similar to the digit string test set. The total number of test set utterances is 4500. The training and test data was pre-processed using the MFCC scheme, where 13-dimensional static coefficients were appended with delta and acceleration coefficients to form observation vectors (Section 2.1). The acoustic model is the cross-word context-dependent triphone hidden Markov model (HMM) with 3 emitting states (Section 2.3.1). The HMM state output distribution is a Gaussian mixture model (GMM) with 12 components and diagonal covariance matrices. The HMM states were tied into 648 physical states using state-level phonetic decision tree clustering (Section 2.4). No language model was used in this task. For digit string recognition, any length digit sequences were allowed. For city name recognition, each of 550 city names was equally possible. The HMM parameters were ML estimated (Section 2.2.3) on the clean training data in a manner similar to [258]. The average word error rate (WER) performance (Section 2.6) of this, clean-trained, HMM system in digit string and city name recognition tasks was 33.55 and 54.68% respectively. The first, HMM, row in Tables 8.1 and 8.2 provides detailed WER performance for each vehicle condition. In order to address the mismatch in noise conditions between the clean training data and the noise-corrupted test sets, the VTS model-based compensation was applied (Section 2.8.2) following the procedure described in [56]. The average WER in these tasks decreased to 2.69 and 14.45% respectively. The second, VTS, row in Tables 8.1 and 8.2 provides detailed WER performance for each vehicle condition.

System	Digit Condition (%)			Average (%)
	IDLE	CITY	HWAY	
HMM	3.85	31.55	65.26	33.55
VTS	1.25	3.09	3.73	2.69

Table 8.1: Clean-trained (HMM) and VTS-compensated (VTS) HMM WER performance in the digit string test set of Toshiba in-car task

System	City Condition (%)		Average (%)
	IDLE	HWAY	
HMM	12.01	97.34	54.68
VTS	6.09	22.81	14.45

Table 8.2: Clean-trained (HMM) and VTS-compensated (VTS) HMM WER performance in the city name test set of Toshiba in-car task

8.1.2 Aurora 4

Aurora 4 is a medium-to-large noise-corrupted speech recognition task [181]. The training set is available in two conditions: clean and multi-style. The clean training data is the WSJ0 subset of WSJ SI284 data [182] consisting of 7138 utterances spoken by 83 speakers and totalling 14 hours of speech (WSJ SI84). The multi-style training data was obtained by artificially corrupting the clean training data using 6 types of noise and two microphone conditions where SNR ranged 10-20 dB. The test set was obtained by artificially corrupting a subset of the development set of 1992 November NIST evaluation [182] using 6 types of noise under two microphone conditions where SNR ranged 5-15 dB. The test set was split into 4 sets: A, B, C and D. The set A contains clean data, set B contains data corrupted by 6 types of noise, set C contains data corrupted by channel distortion and set D contains data corrupted by the noise and channel distortion. The number of utterances in each set is 330, 1980, 330 and 1980 respectively. The training and test data were pre-processed as in Toshiba in-car task. The acoustic model is the cross-word context-dependent triphone HMM with 3 emitting states. The HMM state output distribution is a GMM with 16 components and diagonal covariance matrices. The HMM states were tied into 3143 physical states using state-level phonetic decision tree clustering. A bigram language model with 4988 words in the vocabulary was used. The HMM parameters were ML estimated on the clean training data in a manner similar to Toshiba in-car task. The average WER performance of this, clean-trained, HMM system was 58.47%. The first, HMM, row in Table 8.3 provides detailed WER performance for each test sets. In order to address the mismatch in noise conditions between the clean training data and the noise-corrupted test sets, the VTS model-based compensation was applied following the procedure described in [56]. The second, VTS, row in Table 8.3 shows that the average WER decreased to 17.74%. The HMM parameters were then re-estimated using VTS adaptive training (Section 2.8.2.3) following the procedure described in [56]. The third, VAT, row in Table 8.3 shows that the average WER decreased to 15.93%. The VAT was followed by discriminative VAT based on minimum phone error (MPE) criterion (Section 2.8.2.4). The fourth, DVAT, row in Table 8.3 shows that the average WER decreased to 15.35%.

System	Test Set (%)				Average
	A	B	C	D	(%)
HMM	6.95	55.78	47.28	71.50	58.47
VTS	7.05	15.21	11.89	23.01	17.74
VAT	8.50	13.66	11.81	20.13	15.93
DVAT	7.38	12.91	11.25	19.82	15.35

Table 8.3: Clean-trained (HMM), VTS-compensated (VTS), VTS adaptively trained (VAT) and discriminative VAT (DVAT) HMM WER performance on Aurora 4 task

8.1.3 Aurora 2

Aurora 2 is a small noise-corrupted speech recognition task [183]. The training set is available in two conditions: clean and multi-style. The clean training data consisting of 8440 digit strings up to 7 digits long spoken by 55 male and 55 female US-English speakers. The multi-style training data was obtained by artificially corrupting the clean training data using 4 types of noise where SNR ranged in 5 dB increments: 0, 5, 10, 15 and 20 dB. The test set was obtained by artificially corrupting digit strings spoken by 52 male and 52 female US-English speakers in clean conditions using 8 types of noise where SNR ranged in 5 dB increments: 0, 5, 10, 15 an 20 dB. The test set was split into 3 sets: A, B and C. The set A contains clean data corrupted using the same 4 types of noise as the multi-style training data. The set B contains clean data corrupted using different 4 types of noise. The set C contains half of the clean data corrupted by one type of noise from each set and a channel distortion. The number of utterances in each set is 20002, 20002 and 10001 respectively. The training and test data were pre-processed as in the previous tasks. The acoustic model is the whole-word HMM with 16 emitting states. The HMM state output distribution is a Gaussian mixture model (GMM) with 3 components and diagonal covariance matrices. No language model was used, any length digit sequences were allowed. The HMM parameters were ML estimated on the clean training data in a manner similar to Toshiba in-car task. The average WER performance of this, clean-trained, HMM system was 43.31%. The first, HMM, row in Table 8.4 provides detailed WER performance for each test sets. In order to address the mismatch in noise

System	Test Set (%)			Average (%)
	A	B	C	
HMM	43.86	46.57	35.70	43.31
VTS	9.84	9.11	9.53	9.49
VAT	8.94	8.28	8.79	8.65
DVAT	6.70	6.63	7.04	6.74

Table 8.4: Clean-trained (HMM), VTS-compensated (VTS), VTS adaptively trained (VAT) and discriminative VAT (DVAT) HMM WER performance on the Aurora 2 task averaged over 0-20 dB

conditions between the clean training data and the noise-corrupted test sets, the VTS model-based compensation was applied following the procedure described in [68]. The second, VTS, row in Table 8.4 shows that the average WER decreased to 9.49%. The HMM parameters were then re-estimated using VAT following the procedure described in [56]. The third, VAT, row in Table 8.4 shows that the average WER decreased to 8.65%. The VAT was followed by DVAT analogously to Aurora 4 task. The fourth, DVAT, row in Table 8.4 shows that the average WER decreased to 6.74%.

8.1.4 Software

The work presented in this chapter makes heavy use of several publicly available and proprietary toolkits. Most of the work was conducted using hidden Markov model toolkit (HTK) [265]. The extended acoustic code-breaking part (Chapter 4 and Section 8.2), in addition to HTK, made use of HMM-based speech synthesis system (HTS) [271] and SVM light [109] toolkits. The HTS, an extension of HTK toolkit, was used to artificially generate data. The SVM light toolkit was used to train pair-wise binary SVM classifiers on real and artificially generated data. In addition, an extended version of HTK toolkit was used to estimate and compensate HMM to target noise conditions using vector Taylor series noise compensation approach (see Section 2.8.2). This version of HTK was kindly provided by Toshiba Research Europe Ltd. The conditional augmented model part (Chapter 7 and Section 8.3) was implemented as an extension to the version of HTK toolkit supporting VTS estimation and compensation.

8.2 Extended acoustic code-breaking

There were two sets of experiments performed with extended acoustic code-breaking (Chapter 4). The first set of experiments reported in Section 8.2.1 examined the application to digit string recognition where real training data is available for training the standard, unstructured, discriminative classifiers, such as support vector machines (SVM) discussed in Section 3.2. The previous work has examined and reported positive results on applying the SVM to Aurora 2

[68] and the digit string test set of Toshiba in-car task [67]. For extended acoustic code-breaking, this offers an opportunity to compare SVMs trained on real and artificially generated data. The second set of experiments reported in Section 8.2.2 examined application to city name recognition. The city name test set of Toshiba in-car task, where no training examples of city names exists, offers an opportunity to apply SVMs in the setting not previously possible with these classifiers.

8.2.1 Digit string recognition

For digit string recognition, the experimental setup followed the previous work in [67, 68]. The first variant of acoustic code-breaking in Section 4.1 was implemented. The VTS-compensated HMM was used to produce 1-best hypothesis with segmentation. The segmentation was used to extract observation subsequences with hypothesised word labels. Each observation sub-sequence was then classified into one of digit classes using a discriminative classifier. The discriminative classifier was the SVM implementing the max-wins strategy for multi-class classification (Section 3.2.3.1).

The SVM was trained within the score-space adaptation and compensation framework to yield noise and speaker independent discriminative classifier (Section 6.2.1.3). In order to train the SVM, the multi-style training data was segmented by the VTS-compensated HMM. For each digit pair, an individual SVM was built (Section 3.2.2). For consistency with previous work [68, 278], a subset of multi-style training data comprising 3 out of 4 noise types and 3 out of 5 SNR conditions (10-20 dB) was used. The dynamic kernel associated with the SVM was based on the likelihood ratio score-space $\boldsymbol{\phi}_{\mathbf{r}}$ in equation (6.6), where the VTS-compensated HMM was used to extract features. Table 8.5 provides a short summary of the likelihood ratio score-space.

The extended acoustic code-breaking followed the same approach to training the SVM though artificially generated training data was used to train the SVM. Note that examples of silence were not generated as these are always expected to be available. The artificial data was generated based on the multi-style training data reference transcriptions. There were two synthesis approaches investigated

Notation	Name	Features
$\boldsymbol{\phi}_{\mathtt{r}}$	Likelihood Ratio	log-likelihood ratio all derivatives
$\boldsymbol{\phi}_{\mathtt{l}}$	Likelihood	log-likelihood
$\boldsymbol{\phi}_{\mathtt{l}}^{(1)}$	First-Order Likelihood	log-likelihood all derivatives
$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$	Mean Derivative	log-likelihood mean derivatives
$\boldsymbol{\phi}_{\mathtt{a}}$	Appended Likelihood	all log-likelihoods
$\boldsymbol{\phi}_{\mathtt{m}}$	Matched Context	subset of log-likelihoods

Table 8.5: A summary of score-spaces

for generating the data: HMM synthesis (Section 4.2.1) and statistical HMM synthesis (Section 4.2.2). The initial state-component sequence was obtained based on the inherent HMM state duration densities in equation (2.11). For the HMM synthesis, the observation sequence was sampled at the mean of the multivariate Gaussian distribution associated with the initial state-component sequence. For the statistical HMM synthesis, the static observation sequence was sampled at the static mean of the multivariate Gaussian distribution obtained using the EM algorithm. For both approaches, the VTS-compensated HMM was used to provide the HMM parameters; the VTS transform to be used was drawn randomly.

The first experiment investigated the synthesis approaches on Aurora 2 task, where the whole word acoustic models were used to generate the artificial training data. The WER performance of the SVM trained on real and artificial training data is compared in Table 8.6. The first block of results corresponds to the VTS-compensated HMM (VTS). The second block corresponds to the SVM trained on the real data and the artificial data produced by the HMM synthesis (HMM) and the statistical HMM synthesis (HTS) approaches. The use of dash, —, in the second column corresponds to the use of real training data. The results in Table 8.6 indicate that the VTS-compensated HMM was outperformed by the SVM both trained on the real and artificial training data. The SVM trained on real data showed the best result being on average 19% relatively better. Among

System	Synthesis Approach	Test Set (%)			Avg (%)
		A	B	C	
VTS	—	9.84	9.11	9.53	9.49
SVM	—	7.52	7.35	8.11	7.66
	HMM	9.20	8.51	9.34	9.02
	HTS	8.41	8.03	8.70	8.38

Table 8.6: VTS-compensated HMM (VTS) and SVM restoring WER performance on Aurora 2 task averaged over 0-20 dB, where —, HMM and HTS stand for the use of real (—) and artificial training data generated by HMM synthesis (HMM) and statistical HMM synthesis (HTS)

the synthesis approaches, the statistical HMM synthesis showed the best result achieving on average 61 % of that improvement. The simplest synthesis approach, the HMM synthesis, achieved on average only 25 %. These results indicate that even when the artificial data was produced by the HMM synthesis, which inherits the HMM conditional independence assumptions, a range of discrimination ”clues” was nevertheless carried over to the artificial data which made the SVM possible to correct quarter of the errors corrected based on the real data. The use of more complex synthesis approach, which overcomes the HMM conditional independence assumptions, showed even better results.

The second experiment investigated the statistical HMM synthesis approach on the Toshiba in-car task - a more realistic scenario as the context-dependent acoustic models were used to generate the artificial training data. Note that compared to Aurora 2 task, the test set data is recorded in real noisy environments. The WER performance of the SVM trained on the real and artificial data is compared in Table 8.7. The results in Table 8.7 indicate that the VTS-compensated HMM was outperformed by the SVM both trained on the real and artificial data. The SVM trained on the real data showed the best result being on average 13% relatively better. The SVM trained on the artificial data achieved 47 % of that improvement. These results indicate that even when the context-dependent phone acoustic models were used, a range of discrimination ”clues” was carried over to the artificial data which made the SVM possible to correct almost half of the word errors corrected based on the real data. In addition, these results suggest that the SVM trained on artificial data can be successfully applied to test sets

System	Synthesis Approach	Condition (%)			Avg (%)
		ENON	CITY	HWY	
VTS	—	1.25	3.09	3.73	2.69
SVM	—	1.26	2.60	3.13	2.33
	HTS	1.22	2.88	3.45	2.52

Table 8.7: VTS-compensated HMM (VTS) and SVM restoring WER performance on digit string test set of Toshiba in-car task, where — and HTS stand for the use of real (—) and artificial training data generated by statistical HMM synthesis (HTS)

recorded in real noisy environments.

8.2.2 City name recognition

The previous experimental setup scales quadratically with the number of classes. In order to reduce the computational load for larger vocabulary tasks, it is possible to alter the acoustic code-breaking scheme and/or the discriminative classifier. Following the previous work [128, 246, 248], the acoustic code-breaking scheme was altered whilst keeping the discriminative classifier unchanged. The second variant of acoustic code-breaking was implemented (Section 4.1). The VTS-compensated HMM was used to produce a word lattice. The word lattice was converted into a confusion network. The confusion network was pruned such that each set of parallel arcs contains two confusable city names. The observation sub-sequence was extracted from the earliest start time to the latest end time of the two city names. Each observation sub-sequence was then classified into one or the other city name using the SVM.

The SVM was trained within the score-space adaptation and compensation framework to yield speaker and noise independent discriminative classifier (Section 6.2.1.3). The artificial data was produced based on artificially created reference transcriptions, where the number of samples for each city was set to the average number of samples available to each digit in the above experiments. The HMM parameters for the statistical HMM synthesis were provided by the VTS-compensated HMMs associated with the multi-style training data. A new, random VTS-compensated HMM, which is not associated with the underlying reference transcription, was selected to produce each sample.

The WER performance of the VTS-compensated HMM and SVM trained on the artificial data is compared on the most challenging highway (HWAY) condition in Table 8.8. The relatively high WER performance of the VTS-compensated HMM can be partly attributed to the inherently high perplexity of this test set, where every city name is equally likely. The results in Table 8.8 indicate that the VTS-compensated HMM was outperformed by the SVM trained on the artificial data. The relative improvement obtained on this test set lies above 5 % level consistent with the digit string test set. These results suggest that for larger vocabulary tasks, where the standard acoustic code-breaking is limited to re-scoring only the most frequently confusable word pairs, a larger number of word errors may be corrected by the extended acoustic code-breaking.

System	Synthesis Approach	Condition (%)
		HWAY
VTS	—	22.62
SVM	HTS	21.42

Table 8.8: VTS-compensated HMM (VTS) and SVM restoring WER performance on city name test set of Toshiba in-car task, where HTS stands for the use of artificial training data generated by the statistical HMM synthesis (HTS)

8.2.3 Discussion

This section presented the application of extended acoustic code-breaking to digit string recognition and city name recognition tasks. The experimental results with digit string recognition on Aurora 2 task showed that when the HMM synthesis was used then a range of class discrimination ”clues” was carried over to the artificial training data, which made it possible for the SVM to correct quarter of the errors that were corrected when the SVM was trained on the real data. The use of statistical HMM synthesis, showed even better results by making it possible for the SVM to correct more than half of the errors. The experimental results with digit string recognition on Toshiba in-car data showed that the use of artificial data produced by context-dependent acoustic models adapted to artificially corrupted training data can be used to train the SVM that shows better

WER performance than the VTS-compensated HMM on data recorded in real noisy environments. The experimental results on the city name recognition task showed that consistent with the digit string test set of Toshiba in-car task task 5 % relative WER improvement over the VTS-compensated HMM can be obtained.

The experimental verification in this section was sub-optimal in many ways. First, a medium vocabulary task was used to assess the scheme intended for re-scoring in large vocabulary tasks. Second, the underlying generative model was clean-trained VTS-compensated HMM rather than a discriminative adaptively trained HMM. Third, the use of a more advanced acoustic models for synthesis, such as the trajectory HMM (Section 4.2.2), were not investigated.

There are several aspects that need to be addressed to make the extended acoustic code-breaking useful for re-scoring in large vocabulary tasks. These include efficient, computationally and in terms of the number of samples required, sampling, computationally efficient on-line training and better training data quality providing acoustic models.

8.3 Conditional augmented models

There were two sets of experiments performed with conditional augmented models (Chapter 7). The first set of experiments reported in Section 8.3.1 examined the application of conditional augmented models (CAug) to digit string recognition. The previous work examined and reported positive results on applying monophone/word CAug models to TIMIT [128] and Aurora 2 [278] tasks though different score-spaces, discriminative model parameter estimation criteria and generative models were used. The experiments on Aurora 2 task presented in this section extended the previous work by comparing discriminative model parameter estimation criteria (Section 5.3) and score-spaces (Section 6.2), examining generative model parameter estimation (Section 7.4), inference (Section 5.2.2) and the impact of using adaptively (Section 2.8.2.3) and discriminatively adaptively trained generative models (Section 2.8.2.4). The second set of experiments reported in Section 8.3.2 examined the extension of CAug to medium-to-large vocabulary tasks. The experiments on Aurora 4 task presented in this section examined the use of context-dependent score-spaces, discriminative model parameter tying and adaptively and discriminatively adaptively trained generative models.

8.3.1 Word CAug models

For the lattice re-scoring experiments with word CAug models, the experimental setup followed the previous work in [128, 278]. The VTS-compensated HMM was used to produce a word lattice. For each word arc, the acoustic model score, the HMM log-likelihood, was replaced by the dot-product between the discriminative model parameters and the corresponding score-space feature vector. The 1-best path through the lattice providing the hypothesised word sequence was found using the SCRF/CAug variant of the lattice forward algorithm in equation (5.26), where the summation was replaced by maximum [265]. For the inference experiments with word CAug models, the semi-Markov Viterbi algorithm in equation (5.15) was applied to give the optimal with respect to the discriminative model parameters word sequence. Unless otherwise stated, the WER performance is reported based on the lattice re-scoring.

The CAug was trained within the score-space adaptation and compensation framework to yield noise and speaker independent discriminative model parameters (Section 6.2.1.3). In order to estimated the CAug parameters, the multi-style training data was used. The VTS-compensated HMM was used to produce a pair of numerator, which encodes the reference transcription, and denominator, which encodes a large number of possible transcriptions, word lattice for each training sequence. The numerator lattices contained only the most likely Viterbi segmentation for the reference transcriptions. The denominator lattices contained one or more alignments for each word sequence. The test set A was used as the development set.

The score-spaces examined in this section included the likelihood score-space $\boldsymbol{\phi}_{\mathtt{l}}$, the appended likelihood score-space $\boldsymbol{\phi}_{\mathtt{a}}$ and the mean derivative score-space $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$. Table 8.5 provides a short summary of these score-spaces. The discriminative model parameters associated with all the score-spaces were initialised in the way which guarantees the WER performance of the VTS-compensated HMM (Section 6.2.1.1). For the likelihood score-space, the discriminative model parameters were initialised to unit vector. For the appended likelihood and mean derivative score-space, the discriminative model parameters associated with the log-likelihood feature were initialised to one and the rest to zero. For the silence (`sil`) and short pause (`sp`) classes , the appended likelihood score-space contained only the log-likelihood given the corresponding class.

The rest of this section is organised as follows. Section 8.3.1.1 investigates training criteria for estimating discriminative model parameters using the simplest, likelihood score-space. Section 8.3.1.2 investigates the appended likelihood score-space which incorporates log-likelihoods give all classes rather than one. Section 8.3.1.3 investigates re-estimation of generative models parameters based on CAug using likelihood and appended score-spaces. Section 8.3.1.4 investigates first-order score-spaces based on derivatives of log-likelihood with respect to generative model parameters. Section 8.3.1.5 investigates inference with CAug using the mean derivative score-space compared to the lattice rescoring framework adopted in all previous experiments. Finally, Section 8.3.1.6 examines the use of discriminative and discriminative adaptively trained generative models for extracting features for the use by CAug.

8.3.1.1 Discriminative model parameter estimation

The first experiment investigated estimating discriminative model parameters. The acoustic segment features were provided by the likelihood score-space $\boldsymbol{\phi}_1$ in equation (6.9). There were two parameter estimation criteria examined: CML and MWE (Section 5.3). Compared to the current implementation of large margin training [278], the development time offered by these criteria is more suitable for initial investigation. The regularised versions of these criteria in equations (5.22) and (5.32) were used. The Gaussian prior in equation (3.15), where the mean was set to the initial value of the parameters and covariance was set to the identity matrix scaled by the free parameter σ^{p} in equation (3.16), was used. This provided the guarantee of the WER performance of the VTS-compensated HMM for sufficiently small values of σ^{p}. For both criteria, the gradients were computed in the lattice framework (equations 5.25 and 5.35) and the parameters were optimised using the Rprop algorithm [195].

The optimisation of CML criterion considered multiple values of the free parameter σ^{p} called regularisation constant in the following, where the extreme values 0 and $+\infty$ would effectively turn off the CML objective function and the prior respectively. A summary of the optimisation is given by Figure 8.1, where the left plot shows the change in the CML criterion and the right plot shows the change in the development set WER for σ^{p} ranging from 0 to $+\infty$. Note that the CML criterion was normalised by the number of observations in the training sequences for consistency with the standard practice in MMI estimation of HMM parameters [265]. The optimisation results in Figure 8.1 can be used to make several observations. First, the log-likelihood features provide additional useful for discrimination information. Second, the objective function converges roughly consistently across all values of σ^{p} after 50 iterations. Third, the development set WER performance, apart from the few cases where σ^{p} had large value, indicates good generalisation. Fourth, the suitable range of values for σ^{p} is from 10^{-2} onwards. A range of σ^{p} values, 10^{-1}, ..., $+\infty$ was found optimal with respect to the development set WER. The optimal value was chosen to be $+\infty$ which yields the largest value of the CML criterion.

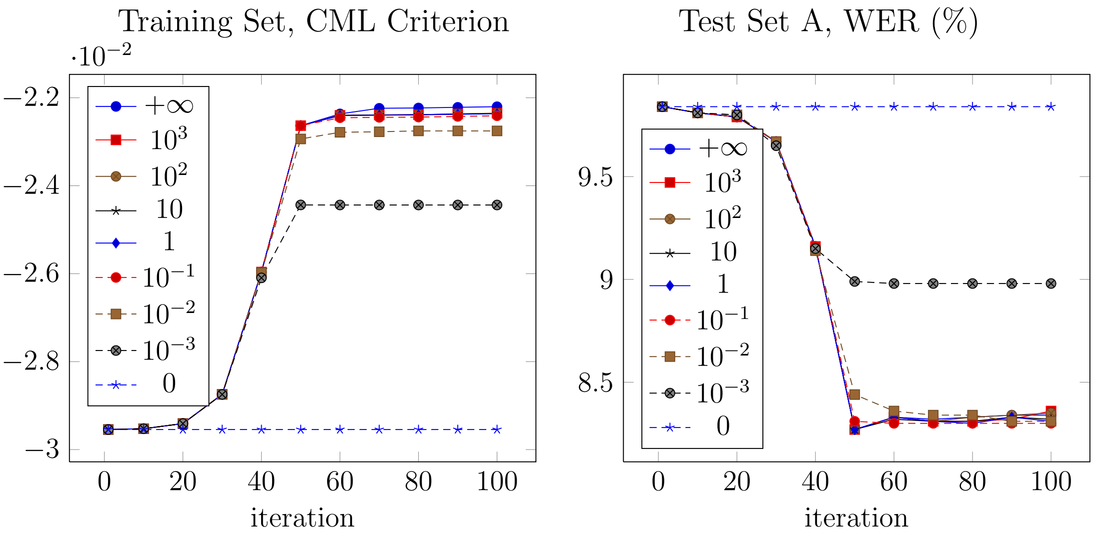


Figure 8.1: Impact of regularisation constant σ^{p} on CML criterion and development set WER in discriminative model parameter estimation for CAug using likelihood score-space $\boldsymbol{\phi}_1$ on Aurora 2 task

The optimisation of MWE criterion followed the CML criterion. A summary of the optimisation is given by Figure 8.2, where the left plot now shows the change in the MWE criterion. Note that the MWE criterion was normalised by the number of words in the reference transcriptions for consistency with the standard practice in MWE/MPE estimation of HMM parameters [265]. The optimisation results in Figure 8.1 can be used to make several observations. First, the log-likelihood features provide additional useful for discrimination information similar to the CML criterion. Second, the objective function requires more iterations to converge for higher values of σ^{p} unlike the CML objective function. Third, the development set WER performance, apart from the few cases where σ^{p} had large value, indicates good generalisation similar to the CML criterion. Fourth, the suitable range of values for σ^{p} is from 10^{-1} onwards unlike for the CML criterion. The optimal with respect to the development set WER value of σ^{p} was 10^{3}.

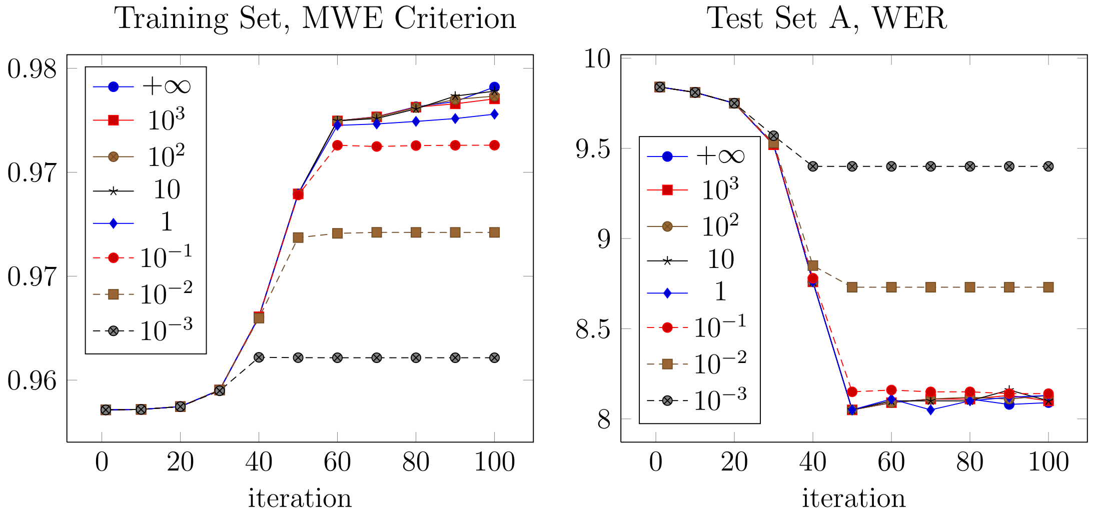


Figure 8.2: Impact of regularisation constant σ^{p} on MWE criterion and development set WER in discriminative model parameter estimation for CAug using likelihood score-space $\boldsymbol{\phi}_1$ on Aurora 2 task

The WER performance of the VTS-compensated HMM and, CML and MWE estimated CAug using the likelihood score-space are compared on Aurora 2 task in Table 8.9. The results in Table 8.9 can be used to make several observations. First, the VTS-compensated HMM was outperformed by the CAug using the likelihood score-space, which essentially introduced HMM-dependent acoustic de-weighting constants (Section 2.7.2.1), on average relatively by 15 % and 18 % respectively. Second, the estimates tuned on the development set show good generalisation on the rest test sets. Third, the use of more complex MWE criterion offers small but consistent improvement over the CML criterion. Since the computation overhead from the use of MWE criterion is marginal, the rest of this chapter will adopt the MWE/MPE criterion.

System	Criterion	Test Set (%)			Average (%)
		A	B	C	
VTS	—	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_1$	CML	8.27	7.79	8.36	8.10
	MWE	8.05	7.44	8.18	7.83

Table 8.9: VTS-compensated HMM (VTS) and CAug ($\boldsymbol{\phi}_1$) lattice restoring WER performance on Aurora 2 task, where $\boldsymbol{\phi}_1$ is likelihood score-space

8.3.1.2 Appended likelihood score-spaces

The CAug using the likelihood score-space $\boldsymbol{\phi}_1$ in Section 8.3.1.1 introduced only one discriminative model parameter for each word. The appended likelihood score-space $\boldsymbol{\phi}_a$ offers the simplest way to increase the number of discriminative model parameters. In addition to the given class log-likelihood, this score-space makes use of log-likelihoods given competing classes.

The second experiment investigated whether the use of competing class log-likelihoods provides additional useful for discrimination information. The optimisation of MWE criterion with CAug using $\boldsymbol{\phi}_a$ followed the approach with $\boldsymbol{\phi}_1$. The lowest WER on the development set, 7.74 %, was achieved when the MWE criterion was equal 0.976 compared to 8.05 % achieved when the MWE criterion was equal to 0.968 with $\boldsymbol{\phi}_1$. This indicates that the use of competing class log-likelihoods provides useful for discrimination information. The optimal value for the regularisation constant σ^{p} was 10^3 which is consistent with $\boldsymbol{\phi}_1$. However, the number of iterations required with $\boldsymbol{\phi}_a$ was 150 compared to 50 with $\boldsymbol{\phi}_1$.

The WER performance of the VTS-compensated HMM and CAug using the likelihood and appended likelihood score-spaces are compared on Aurora 2 task in Table 8.10. The results in Table 8.10 can be used to make several obser-

System	Test Set (%)			Average (%)
	A	B	C	
VTS	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_{\mathsf{l}}$	8.05	7.44	8.18	7.83
$\boldsymbol{\phi}_{\mathsf{a}}$	7.74	7.28	7.94	7.60

Table 8.10: VTS-compensated HMM (VTS) and CAug ($\boldsymbol{\phi}_{\mathsf{l}}$ and $\boldsymbol{\phi}_{\mathsf{a}}$) lattice restoring WER performance on Aurora 2 task, where $\boldsymbol{\phi}_{\mathsf{l}}$ is likelihood and $\boldsymbol{\phi}_{\mathsf{a}}$ is appended likelihood score-space

vations. First, the VTS-compensated HMM was outperformed by CAug using the likelihood and appended likelihood score-spaces on average relatively by 18 % and 20 % respectively. Second, the estimates tuned on the development set showed good generalisation on the rest test sets. Third, the use of competing class log-likelihoods provided additional though small 3 % relative improvement in the WER performance.

8.3.1.3 Generative model parameter estimation

The previous work with CAug in [128, 278] and experiments so far in Section 8.3.1 has considered the use of fixed generative model parameters for parametrising the score-spaces. The experiments so far has considered the use of VTS-compensated HMM. For this form of generative model, it is required to perform a CAug analogue of HMM discriminative VTS adaptive training (DVAT) in Section 2.8.2.4. For the likelihood and appended likelihood score-spaces, the HMM parameter update formulae in the extended Baum-Welch (EBW) form were derived in Section 7.4.3. The statistics used in these update rules was shown to be closely related to the standard statistics. For the likelihood score-space, each lattice arc provided the standard statistics only for one, current class weighted by the corresponding discriminative model parameter. For the appended likelihood score-space, each arc provided the standard statistics for all classes each weighted by the corresponding discriminative model parameter. When the discriminative model parameters are set to their initial values then these update rules yield the standard update rules.

The third experiment investigated the DVAT of HMM parameters for CAug using the likelihood and appended likelihood score-spaces on Aurora 2 task. The experimental setup followed the approach to the CML/MWE estimation of discriminative model parameters in Section 8.3.1.1. The optimisation was performed following the standard approach to the MWE estimation of HMM parameters (Section 2.7.2.2) though the CAug analogue of the standard statistics was accumulated. Note that the standard approach to the DVAT of HMM parameters is based on the VTS-compensated VAT rather than clean-trained HMM [55]. Following the standard approach to the DVAT of HMM parameters (Section 2.8.2.4), the VTS transform parameters were not re-estimated.

The WER performance of the VTS-compensated clean-trained and DVAT HMM, and CAug using the likelihood and appended likelihood score-spaces with estimated discriminative and/or generative model parameters is compared in Table 8.11. The results in Table 8.11 can be used to make several observations.

System	Update		Test Set (%)			Average (%)
	$\boldsymbol{\alpha}$	$\boldsymbol{\lambda}$	A	B	C	
VTS	—	—	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_1$	✗	✓	7.16	6.89	7.52	7.12
	✓	✗	8.06	7.44	8.19	7.84
	✓	✓	6.86	6.61	7.35	6.86
$\boldsymbol{\phi}_{\mathsf{a}}$	✗	✓	7.16	6.89	7.52	7.12
	✓	✗	7.74	7.28	7.94	7.60
	✓	✓	6.84	6.56	7.25	6.81
DVAT	—	—	6.70	6.63	7.04	6.74

Table 8.11: Comparative WER performance on Aurora 2 task for VTS-compensated clean-trained (VTS) and MWE DVAT estimated (DVAT) HMM, and CAug ($\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_{\mathsf{a}}$) with MWE and/or DVAT MWE estimated discriminative and/or generative model parameters, where $\boldsymbol{\phi}_1$ is likelihood and $\boldsymbol{\phi}_{\mathsf{a}}$ is appended likelihood score-space

First, the VTS-compensated clean-trained HMM was outperformed by CAug in every configuration considered. Second, the VTS-compensated DVAT HMM was not outperformed by CAug in any configuration considered. Third, performing the DVAT of HMM given VTS-compensated VAT rather than clean-trained HMM gave on average 5 % relative improvement (lines 2 or 5 and 8). Fourth, for the likelihood and appended likelihood score-spaces estimating generative model parameters on average gave 13 % and 10 % relative improvement. Fifth, the CAug with updated discriminative and generative model parameters using the appended likelihood score-space yielded on average 1 % relative improvement over the likelihood score-space (lines 4 and 7).

8.3.1.4 Mean derivative score-space

The previous sections considered one option to increase the number of features available to each discriminative model class which is to use the log-likelihoods given all classes. A small improvement in the average WER performance was observed on Aurora 2 task over the use of log-likelihood given one class. The experiments reported in this section investigated another option which is to use the derivatives of log-likelihood with respect to generative models parameters. For score-spaces based on the HMM, these derivatives relax the HMM conditional independence assumptions compared to the log-likelihoods (Section 6.2.1.2). In order to avoid potential generalisation issues when using all derivatives (Section 6.2.1.1), a subset consisting of the derivatives with respect to HMM mean vectors was used. The associated score-space in equation (6.11) was called the mean derivative score-space $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$.

The fourth experiment investigated whether the use of derivatives with respect to HMM mean vectors provides additional useful for discrimination information. The experimental setup followed the approach to the CML/MWE estimation of discriminative model parameters in Section 8.3.1.1. The lowest WER on the development set, 7.00 %, was achieved when the MWE criterion was equal to 0.991 compared to 8.05 % and 7.74 % achieved when the MWE criterion was equal to 0.968 and 0.976 with the likelihood $\boldsymbol{\phi}_{\mathtt{l}}$ and appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ score-spaces. This indicated that the use of mean derivatives provides useful for discrimination information. Note that the value of MWE criterion is very high, close to the maximum value of 1. The optimal value for the regularisation constant $\sigma^{\mathtt{p}}$ was $+\infty$ compared to 10^3. This suggests that the use of more complex forms of prior, such as in equation (3.17) where individual regularisation constants are introduced for each dimension, may be advantageous.

The WER performance of the VTS-compensated HMM and CAug using the likelihood $\boldsymbol{\phi}_{\mathtt{l}}$, appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ and mean derivative $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$ score-spaces on Aurora 2 task is compared in Table 8.12. The results in Table 8.12 can be used to make several observations. First, the VTS-compensated HMM was outperformed by CAug using all score-spaces. In particular, the CAug using the mean derivative score-space yielded on average 27 % relative improvement. Second, the CAug using the mean derivative score-space showed good generalisation on the other test sets. Third, the use of mean derivative features yielded on average 8 % relative improvement over the use of log-likelihoods given competing classes.

System	Test Set (%)			Average
	A	B	C	(%)
VTS	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_{\mathtt{l}}$	8.06	7.44	8.19	7.84
$\boldsymbol{\phi}_{\mathtt{a}}$	7.74	7.28	7.94	7.60
$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$	7.00	6.64	7.55	6.97

Table 8.12: Comparative WER performance on Aurora 2 task for VTS-compensated HMM and CAug using likelihood $\boldsymbol{\phi}_{\mathtt{l}}$, appended likelihood $\boldsymbol{\phi}_{\mathtt{a}}$ and mean derivative $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$ score-spaces

8.3.1.5 Inference

The previous sections reported lattice re-scoring WER performance on Aurora 2 task. These lattices contained a subset of the most likely with respect to the VTS-compensated HMM word sequences with one or more possible segmentations. A similar set of lattices was adopted for estimating discriminative model parameters. These segmentations, however, may not be optimal with respect to the CAug (Section 5.2.2). In order to infer word sequences with segmentations optimal with respect to SCRF/CAug, the semi-Markov variant of the Viterbi algorithm in equation (5.15) can be applied.

The fifth experiment investigated the impact of using sub-optimal segmentation on the WER performance on Aurora 2 task. In order to obtain initial, rough estimate, the use of optimal segmentation was investigated in decoding only. Note that the discriminative model parameters in this case remained estimated based on sub-optimal segmentations.

The WER performance of the VTS-compensated HMM and CAug using the mean derivative score-space $\boldsymbol{\phi}_1^{(1,\mu)}$ based on the sub-optimal $\mathbf{a}$ and optimal $\widehat{\mathbf{a}}$ segmentations is compared on Aurora 2 task in Table 8.13. The results in Table 8.13 can be used to make several observations. First, the VTS-compensated HMM was outperformed by the CAug based on the sub-optimal and optimal segmentations. Second, the use of optimal segmentation gave small but consistent improvement in the WER performance on all test sets. This observation suggests that the estimation of discriminative model parameters based on the optimal segmentations may be advantageous.

System	Decoding	Test Set (%)			Average
		A	B	C	(%)
VTS	—	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_1^{(1,\mu)}$	$\mathbf{a}$	7.00	6.64	7.55	6.97
	$\widehat{\mathbf{a}}$	6.78	6.44	7.32	6.75

Table 8.13: Comparative WER performance on Aurora 2 task for VTS-compensated HMM and CAug using mean derivative score-space $\boldsymbol{\phi}_1^{(1,\mu)}$ based on sub-optimal and optimal segmentations in decoding

8.3.1.6 Advanced generative models

The previous section considered the use of VTS-compensated clean-trained HMM to yield features for the mean derivative score-space. The use of more advanced canonical acoustic models, such as the VAT and DVAT HMM, may provide score-spaces with additional useful for discrimination information.

The sixth experiment investigated the use of more advanced canonical acoustic models with CAug using the mean derivative score-space. The experimental setup followed the approach to the CML/MWE estimation of discriminative model parameters in Section 8.3.1.1 though the training and test set lattices were produced by the VTS-compensated VAT and DVAT HMM.

The WER performance of the VTS-compensated VAT and DVAT HMM, and CAug using the mean derivative score-space is compared on Aurora 2 task in Table 8.14. The results in Table 8.14 can be used to make several observations.

System	Test Set (%)			Average (%)
	A	B	C	
VTS	9.84	9.11	9.53	9.49
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	7.00	6.64	7.55	6.97
VAT	8.94	8.28	8.79	8.65
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	6.56	6.53	6.98	6.63
DVAT	6.70	6.63	7.04	6.74
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	6.13	6.21	6.74	6.28

Table 8.14: Comparative WER performance on Aurora 2 task for VTS-compensated clean-trained, VAT and DVAT HMM and corresponding CAug using mean derivative score-space $\boldsymbol{\phi}_{\mathsf{1}}^{(1,\mu)}$

First, the VTS-compensated canonical acoustic models were outperformed by the corresponding CAug using the mean derivative score-space. In particular, the WER performance of the VTS-compensated DVAT HMM was improved relatively by 7 %. Second, the VTS-compensated VAT and DVAT HMM yielded additional useful for discrimination information, which improved the WER performance of the CAug using the mean derivative score-space relatively by 5 % and 10 % respectively, compared to the VTS-compensated clean-trained HMM.

8.3.2 Context-dependent phone CAug models

The previous Section 8.3.1 investigated CAug on Aurora 2 task, where the discriminative acoustic model parameters were associated with individual words. This section investigated CAug on Aurora 4 task, where the discriminative acoustic model parameters are associated with individual context-dependent phones. The experimental setup in this section followed that discussed in Section 8.3.1. The VTS-compensated HMM was used to produce a word lattice. The word lattice was phone-marked to segment each word arc into a sequence phone arcs consistent with the underlying pronunciation. For each phone arc, the acoustic model score, the context-dependent phone HMM log-likelihood, was replaced by the dot-product between the discriminative acoustic model parameters and the corresponding score-space feature vector. The phone arc transitions were set according to equation (5.12) to incorporate the bigram language model and pronunciation probabilities. The 1-best path through the phone-marked lattice providing the hypothesised phone sequence and the associated word sequence was found using the SCRF/CAug variant of the lattice forward algorithm in equation (5.26), where the summation was replaced by maximum [265].

The CAug was trained within the score-space adaptation and compensation framework to yield noise and speaker independent discriminative model parameters (Section 6.2.1.3). The MPE criterion was used to yield estimates. In order to train the CAug, the multi-style training data was used. The VTS-compensated HMM was used to produce a pair of numerator, which encodes the reference transcription with one or more pronunciations, and denominator, which encodes a large number of possible transcriptions with one or more pronunciations, word lattice for each training sequence. The numerator and denominator lattices were phone-marked. The test set B was used as the development set.

The score-spaces examined in this section included the likelihood score-space $\boldsymbol{\phi}_{\mathtt{l}}$, the matched context score-space $\boldsymbol{\phi}_{\mathtt{m}}$ and the mean derivative score-space $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$. Table 8.5 provides a short summary of these score-spaces. The discriminative acoustic model parameters associated with all the score-spaces were initialised in the way which guarantees the WER performance of the VTS-compensated HMM (Section 6.2.1.1). For the likelihood score-space, the discriminative model parameters were initialised to unit vector. For the matched context and mean derivative score-space, the discriminative model parameters associated with the log-likelihood feature given the current class were initialised to one and the rest to zero. For the silence (`sil`) and short pause (`sp`) classes , the matched context score-space contained only the log-likelihood given the corresponding class.

The rest of this section is organised as follows. Section 8.3.2.1 investigates the use of phonetic decision tree clustering proposed in Section 7.3 for tying discriminative acoustic model parameters at the context-dependent phone level using the simplest, likelihood score-space. Section 8.3.2.2 investigates whether the use of matched context and mean derivative score-spaces offers advantages over the likelihood score-space. Section 8.3.2.3 investigates whether for mean derivative score-spaces tying the discriminative acoustic model parameters at the state-level improves robustness with small number of context-dependent phone classes. Section 8.3.2.4 investigates whether the use of more advanced generative models provides additional information useful for discrimination.

8.3.2.1 Phonetic decision tree clustering

The CAug investigated on Aurora 4 task associates discriminative acoustic model parameters with context-dependent phones and generative model, the HMM, parameters with states. A large number of context-dependent phones had limited or no examples in the multi-style training data. In order to address robustness issues when training the CAug, the discriminative acoustic model parameters were tied between context-dependent phones using model-level phonetic decision tree clustering (Section 2.4). Since the HMM parameters were themselves tied using state-level phonetic decision tree clustering then it was important to investigate possible generalisation issues caused by combining the model-level and state-level decision trees in the CAug - the tree intersect effect (Section 7.3).

The experimental setup followed the standard approach to model-level clustering [265]. A single tree was constructed for each possible central phone of all context-dependent phones. Three sets of trees were built with 47 (monophone-level tying), 432 and 4020 leaves. The discriminative acoustic model parameters for seen and unseen context-dependent phones were synthesised by dropping the context-dependent phones down the trees. There were two levels at which clustering was investigated: logical and physical HMM. The logical HMM level referred to clustering all context-dependent phones irrespectively of the state-tied HMM. The physical HMM level referred to clustering only those context-dependent phones which had unique HMM parameters (at least one state is different) associated with them. The latter approach was discussed to be less prone to generalisation issues. The accuracy of both approaches was investigated based on CAug using the simplest likelihood score-space. In order to estimate model-level tied discriminative acoustic model parameters, the MPE criterion was used (Section 5.3). The optimisation procedure followed the approach in Section 8.3.1.1.

The WER performance of the VTS-compensated HMM and CAug using the likelihood score-space based on the two approaches to clustering is compared in Table 8.15. The results in Table 8.15 can be used to make several observations.

System	Clustered HMM Classes	Classes	Test Set (%)				Average (%)
			A	B	C	D	
VTS	—	—	7.05	15.21	11.89	23.01	17.74
ϕ_1	Physical	47	7.57	14.64	11.78	22.22	17.17
		432	7.18	14.34	11.23	22.13	16.95
		4020	6.64	14.24	10.73	21.82	16.70
	Logical	47	7.57	14.64	11.78	22.22	17.17
		432	7.07	14.45	10.86	22.06	16.93
		4020	6.91	14.22	10.76	21.96	16.77

Table 8.15: Model-level phonetic decision tree clustering of physical and logical HMM classes into 47, 432 and 4020 discriminative acoustic model classes. Comparative WER performance on Aurora 4 task for VTS-compensated HMM and CAug using the likelihood score-space $\boldsymbol{\phi}_1$

First, the VTS-compensated HMM was outperformed by CAug in all configurations considered. Second, the use of phonetic decision tree clustering gave small but consistent improvement over monophone-level tying (47 classes). Third, both approaches to clustering show good generalisation as the number of classes increases. Fourth, clustering physical rather than logical HMMs yielded slightly better average WER performance with larger number of classes. For large number of classes, the number of HMMs providing features for each discriminative model class is small. If any of those HMMs are shared among discriminative model classes then it can negatively affect the discriminative model class separability. In order to avoid possible class discrimination issues, the rest of this section will adopt the second approach to clustering.

8.3.2.2 Likelihood, matched context and mean derivative score-spaces

In addition to the likelihood score-space, a range of other score-spaces can be adopted with context-dependent phone CAug (Section 7.2). This section investigated matched-context and mean derivative score-spaces. The matched-context score-space is based on log-likelihoods given context-dependent phones which match the context. For each context-dependent phone the number of such context-dependent phones is equal to the number of monophones. On the other hand, the mean derivative score-space introduces derivatives of log-likelihood with respect to mean vectors. For each context-dependent phone the number of such derivatives equal to the number of components in the associated HMM. The experimental setup followed the approach in Section 8.3.2.1 where 4020 context-dependent phone CAug using the matched-context and mean derivative score-spaces were additionally created.

The WER performance of VTS-compensated HMM and CAug using the likelihood, matched-context and mean derivative score-spaces is compared in Table 8.16 The results in Table 8.16 can be used to make several observations. First, the VTS-compensated HMM was outperformed by CAug using each context-dependent phone score-space. In particular, the use of mean derivative score-space yielded on average 10 % relative improvement in the WER performance. Second, although the use of additional log-likelihoods given context-dependent phones with matched context gave little if any improvement, the distribution of errors across test sets is different which suggests that the use of model combination approaches (Section 2.6.2) may prove successful. Third, the use of mean derivative score-spaces provided additional useful for discrimination information which gave on average small but consistent 4 % relative improvement.

System	Test Set (%)				Average (%)
	A	B	C	D	
VTS	7.05	15.21	11.89	23.01	17.74
$\boldsymbol{\phi}_{\mathtt{l}}$	6.64	14.24	10.73	21.82	16.70
$\boldsymbol{\phi}_{\mathtt{m}}$	6.80	14.21	10.41	21.85	16.69
$\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$	6.70	13.49	10.16	21.11	16.04

Table 8.16: Context-dependent phone score-spaces: likelihood $\boldsymbol{\phi}_{\mathtt{l}}$, matched-context $\boldsymbol{\phi}_{\mathtt{m}}$ and mean derivative $\boldsymbol{\phi}_{\mathtt{l}}^{(1,\mu)}$. Comparative WER performance on Aurora 4 task for VTS-compensated HMM and 4020 context-dependent phone CAug

8.3.2.3 Within-state tied mean derivative score-spaces

When more than one HMM are used to extract features then typically inconsistent order of components within HMM states may negatively affect discrimination across context-dependent phone CAug classes (Section 7.3). This holds for the mean derivative score-space investigated in Section 8.3.2.2, which assumed fixed order of derivatives with respect to state-component mean vectors. The more HMMs are used to extract features the more discrimination using mean derivative score-space is expected to be affected. On the other hand, the likelihood and matched-context score-spaces are not affected since log-likelihoods are not sensitive to the order of components within HMM states. In order to investigate potential class discrimination issues with the mean derivative score-space, the two cases of heavy and light tying were separately investigated based on 47 and 4020 context-dependent phone CAug models. For both cases, the CAug models were built using the mean derivative score-space with mean derivatives summed within states and not.

The WER performance of VTS-compensated HMM and CAug using the likelihood and mean derivative score-space, where mean derivatives are summed within states and not in case of heavy and light tying, is compared in Table 8.17. The results in Table 8.17 can be used to make several observations. First, the VTS-compensated HMM was outperformed by CAug in each case. Second, the CAug using likelihood score-space was outperformed by CAug using the mean derivative score-space in each case. Third, the combination of heavy tying and inconsistent order of HMM components within states indeed affect class discrimination. In

Classes	System	Testing set				Average
		A	B	C	D	
—	VTS	7.05	15.21	11.89	23.01	17.74
47	$\boldsymbol{\phi}_1^{(0)}$	7.57	14.64	11.78	22.22	17.17
	$\boldsymbol{\phi}_1^{(1,\overline{\mu})}$	7.49	14.10	11.31	21.55	16.62
	$\boldsymbol{\phi}_1^{(1,\mu)}$	7.38	14.29	11.71	21.87	16.86
4020	$\boldsymbol{\phi}_1^{(0)}$	6.64	14.24	10.73	21.82	16.70
	$\boldsymbol{\phi}_1^{(1,\overline{\mu})}$	6.82	13.38	10.56	20.57	16.23
	$\boldsymbol{\phi}_1^{(1,\mu)}$	6.70	13.49	10.16	21.11	16.04

Table 8.17: Negative impact of inconsistent component order across HMM states on class discrimination in case of heavy (47 classes) and light (4020 classes) tying with CAug using mean derivative $\boldsymbol{\phi}_1^{(1,\mu)}$ score-space. Usefulness of within-state tied mean derivative $\boldsymbol{\phi}_1^{(1,\overline{\mu})}$ score-space in case of heavy tying. Comparative WER performance on Aurora 4 task for VTS-compensated HMM and 4020 context-dependent phone CAug

particular, the use of mean derivative score-space with mean derivatives summed within HMM states improved the WER performance relatively by 1 %. Fourth, the combination of light tying and inconsistent order of HMM components within states is less prone to class discrimination issues. In particular, the use of mean derivative score-space with mean derivatives summed within HMM states degraded the WER performance relatively by 1 %.

8.3.2.4 Advanced generative models

The experiments with context-dependent phone CAug using mean derivative score-space have so far considered the use of VTS-compensated HMM. The use of more advanced generative models, such as the VAT and DVAT HMM, may provide these score-spaces with additional useful for discrimination information. The experimental setup followed Sections 8.3.1.6 and 8.3.2.2.

The WER performance of VTS-compensated clean-trained, VAT and DVAT HMM and the corresponding 4020 context-dependent phone CAug using the mean derivative score-space is compared in Table 8.18. The results in Table 8.18 can be used to make several observations. First, the CAug gave additional gains over all generative models. Second, the use of VTS-compensated DVAT HMM gave little additional information useful for discrimination resulting in 15.13 % average WER compared to 14.83 % average WER obtained using the VTS-compensated VAT HMM. The most likely explanation to this is over-training of sufficiently

System	Testing set				Average
	A	B	C	D	
VTS	7.05	15.21	11.89	23.01	17.74
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	6.70	13.49	10.16	21.11	16.04
VAT	8.50	13.66	11.81	20.13	15.93
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	7.43	12.57	10.67	19.01	14.83
DVAT	7.38	12.91	11.25	19.82	15.35
$\boldsymbol{\phi}_{\mathsf{b}}^{(1,\mu)}$	6.97	12.66	11.99	19.47	15.13

Table 8.18: Comparative WER performance on Aurora 4 task for VTS-compensated clean-trained, VAT and DVAT HMMs and 4020 context-dependent phone CAug using mean derivative $\boldsymbol{\phi}_{1}^{(1,\mu)}$ score-space

complex HMM on small amount of multi-style training data.

8.3.3 Discussion

This section presented the application of CAug models using scores-spaces based on generative models to noise-corrupted small and medium-to-large vocabulary Aurora 2 and Aurora 4 tasks.

The experimental results on Aurora 2 task with word CAug models gave several indications. First, the use of MWE criterion can offer small but consistent gains over the CML criterion. Second, log-likelihoods for competing words can provide additional useful for discrimination information. Third, re-estimating generative model parameters given estimated discriminative model parameters can give gains. Fourth, derivatives of log-likelihood with respect to mean vectors can provide more additional useful for discrimination information compared to log-likelihoods for competing classes. Fifth, inferring the optimal segmentation of observation sequence into words with respect to CAug rather than relying on segmentation obtained by external classifier can offer small but consistent gains and may yield further gains if performed in training. Sixth, the use of score-spaces based on more complex generative models can give additional useful for discrimination information.

The experimental results on Aurora 4 task with context-dependent phone CAug models gave several indications. First, the use of model-level phonetic decision tree clustering can give small but consistent gains over monophone-level tying. Second, although log-likelihoods for context-dependent phones with matched context provided little if any gain over the likelihood score-space, these can yield complimentary information useful for system combination. Third, derivatives of log-likelihood with respect to mean vectors can provide more additional useful for discrimination information compared to log-likelihoods for context-dependent phones with matched context. Fourth, tying parameters associated with derivatives within HMM states can improve robustness with small number of classes. Fifth, over-trained complex generative models can provide no additional information useful for discrimination compared to simpler models.

The experimental verification in this section was sub-optimal in many ways.

First, only one medium-to-large vocabulary task was used to assess context-dependent phone CAug models. Second, the use of large margin criterion was not investigated. Third, the simplest form of prior was investigated with CML and MWE/MPE criteria. Fourth, only one type of acoustic segment feature-functions was investigated. Fifth, the simplest of supra-segmental feature-functions were used and no attempt was made to estimate the associated parameters.

There are several aspects that need to be addressed to make the context-dependent phone CAug useful for larger vocabulary tasks. These include incorporation of other types of feature-functions, adaptation to speaker and noise conditions with general feature-functions and better optimisation approaches.

8.4 Summary

In this chapter, experimental results with the extended acoustic code-breaking and CAug models were presented on noise-corrupted small, medium and medium-to-large vocabulary tasks. In order to handle the mismatch between noise conditions, both approaches were applied within the score-space adaptation and compensation framework. The VTS model-based compensation was applied. A range of score-spaces were investigated including the likelihood and likelihood ratio score-spaces.

The extended acoustic code-breaking adopted the likelihood ratio score-space. There were two approaches investigated for artificial training data generation: HMM synthesis and statistical HMM synthesis. For digit string recognition on Aurora 2 task, the HMM synthesis approach was found capable of producing observation sequences containing useful for the SVM information to correct 25 % of the errors that can be corrected based on the real training data. The use of more complex statistical HMM synthesis was found to yield even better results - 50 % of the errors were corrected in digit string recognition on Aurora 2 and Toshiba in-car tasks. The results on the city name test set of Toshiba in-car task, where no examples of city names exist in the training data, showed that the use of artificial data trained SVMs can yield gains over the VTS-compensated HMM.

The CAug model adopted score-spaces where features were based on log-likelihoods for one or more classes and derivatives of log-likelihood with respect to generative model parameters. There were a number of aspects investigated on two tasks: Aurora 2 and Aurora 4. The use of CAug models on Aurora 2 task investigated aspects including training criterion, generative model parameter estimation and inference of optimal segmentation. Consistent gains were observed from the use of MWE criterion over the CML criterion, re-estimation of generative model parameters and inferring the optimal segmentation of observation sequence into word sequence with respect to CAug parameters over adopting the segmentations produced by the VTS-compensated HMM. The use of CAug models on Aurora 4 task investigated context-dependent phone score-spaces and parameter tying. Consistent gains were observed from the use of model-level phonetic decision tree clustering to increase the number of context-dependent phone CAug parameters over monophone-level tying. On both tasks the CAug model was found to benefit more from the use of derivatives of log-likelihood with respect to generative model parameters than log-likelihoods for competing classes and perform better than VTS-compensated clean-trained, VAT and DVAT HMMs.

Overall, the concept of using artificial training data for estimating the standard, unstructured discriminative classifiers and the use of structured discriminative classifiers in the form of context-dependent phone CAug models were shown to be promising for larger vocabulary tasks.

Chapter 9

Conclusions

This thesis investigated a discriminative approach to speech recognition based on the standard, unstructured, discriminative models, such as support vector machines (SVM), and structured discriminative models, such as conditional augmented (CAug) models. This thesis contains two major contributions to this area presented in Chapters 4 and 7 and summarised in Sections 9.1 and 9.2.

The first contribution is extended acoustic code-breaking which addresses the limitation of training data insufficiency in the standard acoustic code-breaking schemes by artificially generating examples of under-represented words. This contribution makes it possible to examine application of standard discriminative models to tasks with limited or no examples of the words in the training data, such as city name or large vocabulary speech recognition.

The second contribution is to introduce context-dependent phone CAug models - a structured discriminative model adopting partitioning of sentences into words and words into context-dependent phones. In order to ensure that discriminative acoustic model parameters associated in these models with context-dependent phones are robustly estimated, the use of model-level phonetic decision tree clustering was proposed.

9.1 Extended acoustic code-breaking

One option to handle the vast number of sentences in the discriminative approach to speech recognition is to decompose the whole-sentence recognition problem into a sequence of independent word recognition problems that can be addressed by the standard discriminative models. This serves the basis of a number of acoustic code-breaking schemes discussed in Section 4.1. These schemes have been successfully applied to small and large vocabulary tasks. The current application of these approaches to large vocabulary tasks, however, was limited to re-scoring only a small number of the most frequently occurring word-pair confusions [245]. The main reason for this limitation is the training data insufficiency problem which yields many words with limited or no examples in the training data.

In order to address this issue, the use of artificially generated training data was proposed in Section 4.2 and investigated in Section 8.2. Effectively, a simplified form of speech synthesis is required where observation sequences rather than waveforms are generated. Thus, many of the issues commonly associated in speech synthesis with waveform generation, such as excitation and prosody [166], are not relevant to this approach. Two hidden Markov model (HMM) based approaches examined. The first approach directly used the HMM to generate observation sequences. The procedure is a simple generative process but the generated observation sequences were based on the same conditional independence assumptions as the underlying HMM [72]. In order to overcome the HMM conditional independence assumptions that are often cited as an issue with this simple generative process [75, 270], the second approach applied statistical HMM synthesis [235, 272]. This approach takes into account the deterministic relationship that exists between the static and dynamic parts of observation vectors and produces observation sequences that are not be based on the same conditional independence assumptions as the underlying HMM.

Extended acoustic code-breaking was evaluated on noise-corrupted digit string recognition and city name recognition tasks. There were two digit string recognition tasks investigated: Aurora 2 and digit string test set of Toshiba in-car task. The previous work with the SVM-based acoustic code-breaking reported positive results on both tasks [67, 68]. The first task, Aurora 2, based on whole-word

HMMs was used to compare the two synthesis approaches. It was found that the use of more complex statistical HMM synthesis made it possible for the SVM to correct 35 % more errors than the simplest HMM synthesis. In total, the use of statistical HMM synthesis made it possible for the SVM trained on artificial data to correct 60 % of the errors that were corrected by the SVM trained on real data. The second task, the digit string test set of Toshiba in-car task, based on cross-word triphone HMMs was used to investigate artificial training data generation based on the context-dependent phone rather than whole-word HMMs. It was found that the use of statistical HMM synthesis made it possible for the SVM trained on artificial data to correct approximately 50 % of the errors that were corrected by the SVM trained on real data. The third task, the city name test set of Toshiba in-car task, was used to investigate the extended acoustic code-breaking in the situation, where there is no examples of the words in the training data, to which the standard acoustic code-breaking can not be applied. It was found that the extended acoustic code-breaking gave a small 5 % relative improvement, consistent with digit string test set, over the model-based vector Taylor series (VTS) compensated HMM (Section 2.8.2). Overall, these results showed promise for further investigation of extended acoustic code-breaking.

The major contributions of this part of the thesis are listed below:

(a) sampling observation sequences from hidden Markov models (HMM) compensated to noise conditions using vector Taylor series model-based compensation (Section 4.2.1);

(b) same as above yet taking into account the deterministic relationship between static and dynamic coefficients (Section 4.2.2);

(c) proposing to apply sampling/generation approaches in (a) or (b) with other HMM model-based adaptation/compensation techniques or without;

(d) using (a) and/or (b) possibly with real observation sequences to estimate parameters of support vector machines (SVM) (Section 8.2);

(e) proposing to apply the approach in (d) with other discriminative classifiers;

(f) proposing to use other approaches to sample/generate observation sequences in (d) or (e);

(g) applying (d) in an acoustic code-breaking style experiment (Section 8.2.1);

(h) same as (g) yet in a task where no real examples of words to estimate SVMs for are available (Section 8.2.2).

9.2 Conditional augmented models

As an alternative to acoustic code-breaking, the use of structured discriminative models can be considered to address the vast number of possible sentences. A number of these models were discussed in Chapter 5. These range from maximum entropy models (MEMM), which introduce similar to HMM frame level conditional independence assumptions, to segmental conditional random fields (SCRF), which relax them to the word level. The SCRF, similar to acoustic code-breaking, associates model parameters with individual words. This means that with limited amounts of training data it is hard to ensure robustness of estimates. On the other hand, conditional augmented (CAug) models relax conditional independence assumptions to the phone level, which provides modelling of longer-span dependencies compared to the HMM and MEMM, and better coverage in the training data over the SCRF. The previous work reported positive results on two small vocabulary tasks, Aurora 2 and TIMIT, based on word [278] and monophone [128] CAug models.

In order to make CAug models more generally applicable, the two directions of word and phone level modelling need to be combined. The context-dependent phone CAug model proposed in Chapter 7 applies word-level conditional independence assumptions to extract information from word and pronunciation sequences (Section 6.3), and phone-level conditional independence assumptions to extract information from observations sequences (Section 6.2). In order to ensure that parameters associated with context-dependent phones are robustly estimated, the use of model-level phonetic decision tree clustering was proposed to automatically balance the complexity of CAug model against the amount of training data.

The use of acoustic segment feature-functions (Section 6.2) based on generative models to extract features, such as likelihood, appended likelihood and mean derivative score-spaces (Section 6.2.1.1), introduces generative model parameters into the set of CAug model parameters. There are several advantages of using these feature-functions, such as the availability of systematic approaches to introduce more features, and giving opportunities for training speaker and noise independent discriminative models by adapting generative model parameters using model-based adaptation/compensation techniques. In the previous work, the generative model, the HMM, parameters were assumed to be given and fixed. This work derived update rules in the extended Baum-Welch form for re-estimating HMM parameters based on conditional maximum likelihood (CML) and minimum word/phone error criteria for CAug models using the likelihood and appended likelihood score-spaces (Section 7.4). In addition to the standard discriminative parameter estimation, the update rules were derived for discriminative (DSAT) speaker adaptive training based on constrained maximum likelihood linear regression (CMLLR), CMLLR-based DSAT, and discriminative VTS adaptive training (DVAT).

The context-dependent phone CAug models were evaluated on two noise-corrupted tasks: small vocabulary Aurora 2 and medium-to-large vocabulary Aurora 4. The first task, Aurora 2, was used to investigate the various options available with this structured discriminative model such as parameter estimation criteria, inference of optimal segmentation of observation sequences into words, feature-functions, and re-estimation of generative model parameters for those feature-functions based on generative models. The experimental results in Section 8.3.1 gave several indications which were summarised in Section 8.3.3. In particular, it was found that the use of mean derivative over likelihood and appended likelihood score-spaces gave consistent gains as well as re-estimating over fixing the HMM parameters in the likelihood and appended likelihood score-spaces. The second task, Aurora 4, was used to investigate the context-dependent phone CAug model on a medium-to-large vocabulary task. The experimental results in Section 8.3.2 gave several indications which have been summarised in Section 8.3.3. In particular, the use of model-level phonetic decision tree clustering over monophone-level tying gave consistent gains as the use of mean derivative over likelihood and matched context score-spaces. Overall, these results showed promise for further investigation of context-dependent phone CAug models.

The major contributions of this part of the thesis are listed below:

(a) conditional augmented (CAug) model where acoustic model parameters/classes are defined at the context-dependent phone level and language and pronunciation model parameters/classes are defined at the word level (Section 7.1);

(b) parameter tying of context-dependent phone CAug classes based on model-level phonetic decision trees (Section 7.3);

(c) same as above yet parameters of context-dependent phone CAug classes are ties at the monophone level (Section 8.3.2.1);

(d) investigating the use of likelihood, matched-context and mean derivative score-spaces based on context-dependent hidden Markov models (HMM) adapted to noise using vector Taylor series (VTS) approach (Section 7.2);

(e) deriving conditional maximum likelihood and minimum phone/word error update rules in the extended Baum-Welch form for estimating HMM parameters of CAug models based on likelihood and appended all score-spaces (Section 7.4);

(f) same as above yet canonical HMM parameters are adapted to speaker conditions using constrained maximum likelihood linear regression (Section 7.4);

(g) same as above yet canonical HMM parameters are adapted to noise conditions using VTS (Section 7.4);

(h) verifying the above experimentally in noise-corrupted small and medium-to-large vocabulary speech recognition tasks (Sections 8.3.1 and 8.3.2).

9.3 Future work

There are several aspects presented in this thesis that may benefit from further investigation. A number of suggestions are given below.

For the extended acoustic code-breaking, it may be worth investigating the following aspects.

- The acoustic model used for generating artificial training data was the HMM. More advanced acoustic models such as the trajectory HMM, and synthesis approaches such as those using global variance, may yield artificial data of better quality (Section 4.2.2).
- The discriminative model used for the experiments in Section 8.2.1 was the SVM implementing multi-class classification using max-wins strategy (Section 3.2.3.1). The use of direct multi-class discriminative models, such as the MaxEnt models, may be advantageous. In addition, the use of more powerful feature-functions over the likelihood ratio score-space based on the HMM (Section 6.2) adopted in this work may be advantageous.
- The evaluation presented in this thesis considered a medium vocabulary task. An application to larger tasks are required to give a fair assessment of the approach intended to be used for large vocabulary tasks.
- Finally, the need for efficient (computationally and in terms of the number of samples required) sampling and computationally efficient on-line discriminative model training must be addressed to make this approach practically useful for re-scoring in speech recognition tasks.

For the context-dependent phone CAug models, it may be worth investigating the following aspects.

- The acoustic segment feature-functions, which were investigated in Section 8.3, included zero- and first-order score-spaces, such as the likelihood, appended-likelihood, and mean derivative score-spaces. The second-order score-spaces as discussed in Section 6.2.1.2 offer an opportunity to model more complex dependencies, which may benefit the CAug model. In addition, the use of alternative to HMM generative models, such as the trajectory HMM (Section 4.2.2), may yield even more powerful features.
- The use of supra-segmental features, which were investigated in Section 8.3.2, were limited to n-gram language and pronunciation model log-probabilities.

The use of other supra-segmental feature-functions, as discussed in Section 6.3, can provide additional information.

- The parameter estimation criteria, which were investigated in Section 8.3.1, included CML and MWE/MPE. The use of margin-based criteria, such as the perceptron (Section 2.7.1.5) and large margin (Section 5.3.3), may be advantageous. In addition, the use of kernelisation (Section 3.2) may help to address computational issues associated with high-dimensional feature-spaces.

- The inference of optimal segmentation of observation sequences into words were investigated only in decoding in Section 8.3.1.5. The use of optimal segmentation, also in training, may be advantageous (Section 5.2.2). In order to extend this approach to larger vocabulary tasks, it is important to address high complexity of semi-Markov Viterbi algorithm. For complex feature-functions, such as high-order score-spaces, this is particularly important.

- The optimisation approach used for the experiments in Section 8.3 was the Rprop algorithm, which required on average more than 50 iterations to converge. Availability of better optimisation approaches converging in fewer iterations would be an advantage.

Appendices

Appendix A
Conditional augmented model parameter estimation

A.1 Gradients of the CML objective function

The CML objective function is given by

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log(P(\mathbf{w}_{1:L_r}^{(r)}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})) \tag{1}$$

where $\boldsymbol{\alpha}$ are the discriminative model parameters, $\boldsymbol{\lambda}$ are the generative model parameters and $P(\cdot)$ is the CAug posterior given by equation (7.1). Substituting the CAug posterior into $\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D})$ gives

$$\begin{aligned}\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w}_{1:L_r}^{(r)};\boldsymbol{\lambda}))\right)- \\ \log\left(\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\right)\end{aligned} \tag{2}$$

The CML objective function can be split into two parts

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) - \mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) \tag{3}$$

where

$$\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w}^{(r)}_{1:L_r};\boldsymbol{\lambda}))\right) \tag{4}$$

is the numerator term objective function and

$$\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\right) \tag{5}$$

is the denominator term objective function. The gradients with respect to discriminative $\boldsymbol{\alpha}$ and generative $\boldsymbol{\lambda}$ model parameters are derived in Sections A.1.1 and A.1.2 respectively.

A.1.1 Gradient with respect to discriminative model parameters

The gradient of $\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D})$ with respect to $\boldsymbol{\alpha}$ is given by

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) - \nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) \tag{6}$$

where the gradient of $\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D})$ with respect to $\boldsymbol{\lambda}$ can be computed as follows

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) =$$

$$= \frac{1}{R}\sum_{r=1}^{R}\frac{\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\nabla_{\boldsymbol{\alpha}}\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda})\}}{\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))} = \tag{7}$$

$$= \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}}P(\mathbf{a},\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha},\boldsymbol{\lambda})\nabla_{\boldsymbol{\alpha}}\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda})\} = \tag{8}$$

$$= \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}}P(\mathbf{a},\mathbf{w}|\mathbf{O}^{(r)}_{1:T_r};\boldsymbol{\alpha},\boldsymbol{\lambda})\boldsymbol{\phi}(\mathbf{O}^{(r)}_{1:T_r},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}) \tag{9}$$

and, following the same approach as above, the gradient of $\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\alpha}$ is given by

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}; \boldsymbol{\lambda}) \quad (10)$$

Thus

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\Bigg[\sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}; \boldsymbol{\lambda}) - \sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda})\Bigg] \quad (11)$$

A.1.2 Gradient with respect to generative model parameters

The gradient of $\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ is given by

$$\nabla_{\boldsymbol{\lambda}}\mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \nabla_{\boldsymbol{\lambda}}\mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) - \nabla_{\boldsymbol{\lambda}}\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) \quad (12)$$

where the gradient of $\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ can be computed as follows

$$\begin{aligned}
&\nabla_{\boldsymbol{\lambda}}\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \\
&= \frac{1}{R}\sum_{r=1}^{R}\frac{\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda})\}}{\sum_{\mathbf{w}}\sum_{\mathbf{a}}\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))} = && (13)\\
&= \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda})\} = && (14)\\
&= \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\sum_{s=1}^{|\mathbf{a}|}\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} && (15)
\end{aligned}$$

and, following the same approach as above, the gradient of $\mathcal{F}_{\texttt{den}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ is given by

$$\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\texttt{num}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \nabla_{\boldsymbol{\lambda}} \{\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} \tag{16}$$

Thus

$$\begin{aligned} &\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \\ &= \frac{1}{R} \sum_{r=1}^{R} \Bigg[\sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \nabla_{\boldsymbol{\lambda}} \{\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} - \\ &\qquad \sum_{\mathbf{w}} \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \nabla_{\boldsymbol{\lambda}} \{\boldsymbol{\alpha}_{\texttt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}^{(r)}, a_s; \boldsymbol{\lambda})\} \Bigg] \end{aligned} \tag{17}$$

For example, when the acoustic segment features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda})$ are provided by the zero-order likelihood score-space in equation (6.9) then the gradient above becomes

$$\begin{aligned} &\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\texttt{cml}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \\ &= \frac{1}{R} \sum_{r=1}^{R} \Bigg[\sum_{\mathbf{a}} P(\mathbf{a}|\mathbf{w}_{1:L_r}^{(r)}, \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \alpha_{\texttt{am}}^{(a_s^{\texttt{i}})} \nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}}^{(r)}|a_s^{\texttt{i}}; \boldsymbol{\lambda})) - \\ &\qquad \sum_{\mathbf{w}} \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \sum_{s=1}^{|\mathbf{a}|} \alpha_{\texttt{am}}^{(a_s^{\texttt{i}})} \nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}}^{(r)}|a_s^{\texttt{i}}; \boldsymbol{\lambda})) \Bigg] \end{aligned} \tag{18}$$

A.2 Gradients of the MBR objective function

The MBR objective function is given by

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \mathcal{L}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)}) \tag{19}$$

The variant based on accuracy function is adopted with CAug

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{A}(\mathbf{w}, \mathbf{w}_{1:L_r}^{(r)}) \tag{20}$$

In this work the accuracy function is defined at the level of generative models: word or phone. The objective function to maximise for minimum word/phone error (MWE/MPE) estimation of generative model parameters is given by

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{w}, \mathbf{a}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) \tag{21}$$

The gradients with respect to discriminative $\boldsymbol{\alpha}$ and generative $\boldsymbol{\lambda}$ model parameters are derived in Sections A.2.1 and A.2.2 respectively.

A.2.1 Gradient with respect to discriminative model parameters

The gradient of $\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\alpha}$ is given by

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}\sum_{\mathbf{a}} \nabla_{\boldsymbol{\alpha}}\{P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\}\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) \tag{22}$$

where

$$\nabla_{\boldsymbol{\alpha}}\{P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})\} = \nabla_{\boldsymbol{\alpha}}\left\{\frac{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))}{Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})}\right\} = \tag{23}$$

$$= \frac{\nabla_{\boldsymbol{\alpha}} \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})}{Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})^2} - \tag{24}$$

$$\frac{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}))\nabla_{\boldsymbol{\alpha}}Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})}{Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})^2} = \tag{25}$$

$$= P(\mathbf{a}, \mathbf{w}|\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda})(\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)}, \mathbf{a}, \mathbf{w}; \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\alpha}} \log(Z(\mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}))) \tag{26}$$

The gradient of $\log(Z(\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda}))$ with respect to $\boldsymbol{\alpha}$ is

$$\nabla_{\boldsymbol{\alpha}} \log(Z(\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})) = \frac{\sum_{\mathbf{w}}\sum_{\mathbf{a}} \nabla_{\boldsymbol{\alpha}}\{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\}}{Z(\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})} = \tag{27}$$

$$= \sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}) \tag{28}$$

The gradient of $\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D})$ with respect to $\boldsymbol{\alpha}$ then becomes

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D}) = \frac{1}{R}\sum_{r=1}^{R}\Bigg[\sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})\Bigg(\mathcal{A}(\mathbf{a},\mathbf{w}_{1:L_r}^{(r)}) - \tag{29}$$

$$\sum_{\mathbf{w}'}\sum_{\mathbf{a}'} P(\mathbf{a}',\mathbf{w}'|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})\mathcal{A}(\mathbf{a}',\mathbf{w}_{1:L_r}^{(r)})\Bigg)\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda})\Bigg] \tag{30}$$

A.2.2 Gradient with respect to generative model parameters

The gradient of $\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha},\boldsymbol{\lambda};\mathcal{D})$ with respect to $\boldsymbol{\lambda}$ can be obtained following the derivation given in the previous section where, however, the gradients are computed with respect to generative rather than discriminative model parameters. In particular,

$$\nabla_{\boldsymbol{\lambda}}\{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\} =$$

$$= \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\sum_{s=1}^{|\mathbf{a}|}\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}},a_s;\boldsymbol{\lambda})\} \tag{31}$$

and

$$\nabla_{\boldsymbol{\lambda}}\log(Z(\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})) =$$

$$= \frac{\sum_{\mathbf{w}}\sum_{\mathbf{a}} \nabla_{\boldsymbol{\lambda}}\{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:T_r}^{(r)},\mathbf{a},\mathbf{w};\boldsymbol{\lambda}))\}}{Z(\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})} = \tag{32}$$

$$= \sum_{\mathbf{w}}\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{w}|\mathbf{O}_{1:T_r}^{(r)};\boldsymbol{\alpha},\boldsymbol{\lambda})\sum_{s=1}^{|\mathbf{a}|}\nabla_{\boldsymbol{\lambda}}\{\boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}},a_s;\boldsymbol{\lambda})\} \tag{33}$$

From which the gradient of $\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D})$ with respect to $\boldsymbol{\lambda}$ is given by

$$\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \left[\sum_{\mathbf{w}} \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w} | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \left(\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) - \right.\right.$$
$$\left.\left. \sum_{\mathbf{w}'} \sum_{\mathbf{a}'} P(\mathbf{a}', \mathbf{w}' | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \mathcal{A}(\mathbf{a}', \mathbf{w}_{1:L_r}^{(r)}) \right) \sum_{s=1}^{|\mathbf{a}|} \nabla_{\boldsymbol{\lambda}} \{ \boldsymbol{\alpha}_{\mathtt{am}}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda}) \} \right] \quad (34)$$

For example, when the acoustic segment features $\boldsymbol{\phi}(\mathbf{O}_{\{a_s\}}, a_s; \boldsymbol{\lambda})$ are provided by the zero-order likelihood score-space in equation (6.9) then the gradient above becomes

$$\nabla_{\boldsymbol{\lambda}} \mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}; \mathcal{D}) = \frac{1}{R} \sum_{r=1}^{R} \left[\sum_{\mathbf{w}} \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{w} | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \left(\mathcal{A}(\mathbf{a}, \mathbf{w}_{1:L_r}^{(r)}) - \right.\right.$$
$$\left.\left. \sum_{\mathbf{w}'} \sum_{\mathbf{a}'} P(\mathbf{a}', \mathbf{w}' | \mathbf{O}_{1:T_r}^{(r)}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \mathcal{A}(\mathbf{a}', \mathbf{w}_{1:L_r}^{(r)}) \right) \sum_{s=1}^{|\mathbf{a}|} \alpha_{\mathtt{am}}^{(a_s^{\mathtt{i}})} \nabla_{\boldsymbol{\lambda}} \log(p(\mathbf{O}_{\{a_s\}}^{(r)} | a_s^{\mathtt{i}}; \boldsymbol{\lambda}) \right] \quad (35)$$

References

[1] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Kluwer Academic Publishers, 1993. 49, 57

[2] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. HMM Adaptation using Vector Taylor Series for Noisy Speech Recognition. In *Proceedings of the sixth international conference on spoken language processing*, **3**, pages 869–872, 2000. 49, 57, 58, 59

[3] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, **25**:821–837, 1964. 82

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proceedings of the fourth international conference on spoken language processing*, pages 1137–1140, Philadelphia, PA, USA, 1996. 52

[5] E. Arisoy, M. Saraclar, B. Roark, and I. Shafran. Syntactic and sub-lexical features for Turkish discriminative language models. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 5538–5541, 2010. 130, 131

[6] E. Arisoy, M. Saraclar, B. Roark, and I. Shafran. Discriminative language modeling with linguistic and statistically derived features. *IEEE Transactions on Audio, Speech and Language Processing*, **20**[2]:540–550, 2012. 130, 131

[7] S. Axelrod, V. Goel, V. Gopinath, P. Olsen, and K. Visweswariah. Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, **15**[1]:172–189, 2007. 41

[8] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 49–52, Tokyo, Japan, 1986. 19, 36, 41

[9] L. R. Bahl, P. V. de Souza, P. S. Gopalkrishnan, D. Nahamoo, and M. A. Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proceedings of DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991. 23

[10] R. Bakis. Continuous speech word recognition via centisecond acoustic states. *Journal of the Acoustical Society of America*, **59**:S97–S97, 1976. 13

[11] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, **73**:360–363, 1967. 16

[12] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**:164–171, 1970. 11, 19, 20, 42

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, **3**:1137–1155, 2003. 30

[14] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**:39–71, 1996. 69, 70, 71, 73

[15] J. A. Bilmes. Graphical models and automatic speech recognition. In M. Ostendorf M. Johnson, S. Khudanpur and R. Rosenfield, editors, *Mathematical Foundations of Speech and Language Processing*, **138**, pages 191–245. Springer, New York, 2004. 13, 29

[16] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006. 40

[17] C. M. Bishop and J. Lasserre. Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, **8**:3–24, 2007. 1, 2

[18] S. Boll. Suppression of acoustic noise in speech using spectral subtractio. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **27**:113–120, 1979. 78

[19] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of fifth annual workshop on computational learning theory*, pages 144–152, 1992. 81, 82

[20] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: A case study in handwritting digit recognition. In *Proceedings of the international conference on pattern recognition*, **2**, pages 77–82, Jerusalem, Israel, 1994. 84, 85

[21] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2009. 80

[22] P. F. Brown. *The acoustic-modelling problem in automatic speech recognition*. PhD thesis, Carnegie Mellon University, 1987. 35

[23] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19**[2]:263–311, 1993. 22

[24] P. F. Brown, V. J. Della Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, **18**[4]:467–479, 1992. 30

[25] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**:121–167, 1998. 79, 80, 82

[26] W. J. Byrne. Minimum bayes risk estimation and decoding in large vocabulary continuous speech recognition. *IEICE Transactions on Information and Systems: Special Issue on Statistical Modelling for Speech Recognition*, **E89-D**[3]:900–907, 2006. 3, 33, 37

[27] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal processing letters*, **13**:308–311, 2006. 122

[28] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech and Language*, **20**:382–399, 2006. 77

[29] C. Chelba and F. Jelinek. Structured language modelling. *Computer Speech and Language*, **14**:283–332, 2000. 131

[30] S. F. Chen. Performance prediction for exponential language models. In *Proceeding of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 450–458, 2009. 75

[31] S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modelling. Technical Report TR-10-98, Harvard University, 1998. 29, 30

[32] S. F. Chen and R. Rosenfeld. Efficient sampling and feature selection in whole sentence maximum entropy language models. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **1**, pages 549–552, Phoenix, AZ, USA, 1999. 130, 131

[33] S. F. Chen and R. Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, **8**:37–50, 2000. 74

[34] S. S. Chen and R. A. Gopinath. Model selection in acoustic modelling. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1087–1090, Rhodes, Greece, 1997. 13

[35] W. Chou, C. H. Lee, and B. H. Juang. Minimum error rate training based on n-best string models. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 652–655, Minneapolis, USA, 1993. 37

[36] M. Collins, B. Roark, and M. Saraclar. Discriminative syntactic language modelling for speech recognition. In *Proceedings of the forty-third annual meeting of the association for computational linguistics*, pages 507–514, 2005. 130, 131

[37] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, **20**:273–297, 1995. 78, 79, 80, 81, 82

[38] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. In N. Cristianini, J. Shawe-Taylor, and B. Williamson, editors, *Journal of machine learning research*, **2**, pages 265–292. MIT Press, 2001. 84, 85, 86

[39] R. B. D'Agostino and H. K. Russell. *Factor Loading Matrix*. John Wiley & Sons, Ltd, 2005. Encyclopedia of Biostatistics. 61

[40] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, **43**:1470–1480, 1972. 73

[41] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Speech and Audio Processing*, **28**[4]:357–366, 1980. 8, 9

[42] M. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970. 44

[43] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison-Wesley, 3rd edition, 2002. 84

[44] S. A. Della Pietra, V. J. Della Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**:380–393, 1997. 73

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**:1–39, 1977. 20, 42

[46] L. Deng, A. Acero, M. Plumpe, and X. D. Huang. Large-vocabulary speech recognition under adverse acoustic environments. In *Proceedings of the sixth international conference on spoken language processing*, **3**, pages 806–809, Beijing, China, 2000. 78

[47] A. Deoras, D. Filimonov, M. Harper, and F. Jelinek. Model combination for speech recognition using empirical Bayes risk minimization. In *Proceeding of IEEE Workshop on Spoken Language Technology*, pages 235–240, Berkeley, CA, USA, 2010. 76

[48] D. Doermann and K Tombre (editors). *Handbook of Document Image Processing and Recognition*. Springer, 2014. Lecture notes in computer science. 1

[49] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001. 16, 18, 40, 77, 107, 108

[50] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa. A support vector machine approach for detection of microcalcifications. *IEEE Transactions on medical imaging*, **21**[12]:1552–1563, 2002. 78

[51] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland, and K. Yu. Training LVCSR systems on thousands of hours of data. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 209–212, 2005. 91

[52] G. Evermann and P. C. Woodland. Posterior probability decoding, confidence estimation and system combination. In *Proceedings of NIST Speech Transcription Workshop*, Baltimore, USA, 2000. 3, 32, 33

[53] M. Ferras, C. C. Leung, C. Barras, and J.-L. Gauvain. Constrained MLLR for speaker recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages IV–53–IV–56, 2007. 122

[54] J. Fiscus. A post-processing system to yield reduced word error rates: Recog- niser output voting error reduction (ROVER). In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 347–352, Santa Barbara, USA, 1997. 3, 33

[55] F. Flego and M. J. F. Gales. Discriminative adaptive training with VTS and JUD. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 170–175, Merano, Italy, 2009. 42, 44, 55, 60, 65, 66, 151, 173

[56] F. Flego and M. J. F. Gales. Adaptive training and noise estimation for model-based noise compensation for ASR. Technical Report CUED/F-INFENG/TR653, University of Cambridge, 2010. 157, 158, 160

[57] J. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, 1996. 85

[58] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **29**:254–272, 1981. 1, 78

[59] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **34**:52–59, 1986. 10

[60] S. Furui. Robust methods in automatic speech recognition and understanding. In *Proceedings of the eighth European conference on speech communication and technology*, pages 1993–1998, 2003. 1

[61] M. J. F. Gales. *Model-based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 1995. 19, 49, 57, 59

[62] M. J. F. Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996. 55, 56

[63] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, **12**[2]:75–98, 1998. 49, 50, 51, 52, 53, 54

[64] M. J. F. Gales. Discriminative models for speech recognition. In *Proceedings of Information Theory and Applications Workshop*, pages 170–176, University of California, San Diego, USA, February 2007. 36, 38, 39, 40, 127, 128, 129

[65] M. J. F. Gales. Model-based approaches to handling uncertainty. In D. Kolossa and R. Haeb-Umbach, editors, *Robust speech recognition of uncertain or missing data. Theory and applications*, chapter 5, pages 101–126. Springer, 2011. 49, 59, 60, 63, 65, 66, 151

[66] M. J. F. Gales and F. Felgo. Theory for model-based noise compensation schemes. Technical Report CUED/F-INFENG/TR???, Cambridge University, 2012. 60, 61, 62, 63, 64, 65, 66, 151

[67] M. J. F. Gales and F. Flego. Combining VTS model compensation and support vector machines. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 3821–3824, 2009. 161, 188

[68] M. J. F. Gales and F. Flego. Discriminative classifiers with adaptive kernels for noise robust speech recognition. *Computer Speech and Language*, **24**:648–662, 2010. 3, 69, 78, 84, 85, 87, 89, 90, 91, 129, 160, 161, 188

[69] M. J. F. Gales, P. C. Kim, D. Y. Woodland, D. Mrva, R. Sinha, and S. E. Tranter. Progress in the cu-htk broadcast news transcription

system. *IEEE Transactions on Speech and Audio Processing*, **14**[5]:1513–1525, 2006. 13

[70] M. J. F. Gales and M. I. Layton. SVMs, score-spaces and maximum margin statistical models. In *Proceedings of Workshop on statistical modeling approach for speech recognition*, Kyoto, Japan, December 2004. 2

[71] M. J. F. Gales and C. Longworth. Discriminative classifiers with generative kernels for noise robust ASR. In *Proceedings of the ninth Annual Conference of the International Speech Communication Association*, Brisbane, Australia, 2008. 87

[72] M. J. F. Gales, A. Ragni, H. AlDamarki, and C. Gautier. Support vector machines for noise robust ASR. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 205–210, 2009. i, 2, 78, 84, 89, 91, 92, 94, 188

[73] M. J. F. Gales, S. Watanabe, and E. Fosler-Lussier. Structured discriminative models for speech recognition. *IEEE Signal Processing Magazine. Special issue on fundamental technologies in modern speech recognition*, pages 70–81, 2012. 2, 3, 7, 11, 22, 29, 36, 40, 69, 71, 72, 73, 77, 78, 85, 87, 88, 89, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 112, 113, 115, 116, 119, 120, 121, 122, 126, 127, 130, 133, 148

[74] M. J. F. Gales and P. C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, **10**:249–264, 1996. 49, 56

[75] M. J. F. Gales and S. J. Young. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, **1**[3]:195–304, 2007. 2, 3, 7, 9, 10, 11, 13, 14, 15, 19, 21, 22, 23, 25, 26, 29, 30, 31, 32, 33, 35, 36, 37, 38, 41, 43, 45, 49, 50, 51, 52, 53, 55, 56, 57, 58, 77, 87, 95, 96, 97, 188

[76] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, **2**:291–298, 1994. 49

[77] Z. Ghahramani. Learning dynamic Bayesian networks. In C. L. Giles and M. Gori, editors, *Adaptive processing of temporal information. Lecture notes in artificial intelligence*, **1387**, pages 168–197. Springer Verlag, 1997. 13, 29

[78] A. Globerson, T. Y. Koo, X. Carreras, and M. Collins. Exponentiated gradient algorithms for log-linear structured prediction. In *Proceedings of the twenty fourth international conference on Machine learning*, pages 305–312, 2007. 41

[79] I. J. Good. The population frequency of species and the estimation of population parameters. *Biometrika*, **40**:237–264, 1953. 30

[80] J. Goodman. Exponential priors for maximum entropy models. In *Proceeding of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 305–312, Boston, MA, USA, 2004. 74, 75

[81] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo. An inequality for rational fractions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, **37**[1], 1991. 41

[82] R. A. Gopinath, M. J. F. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny. Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task. In *Proceedings of ARPA Workshop on Spoken Language System Technology*, pages 127–130, Austin, Texas, USA, 1995. 59

[83] R. A. Gopinath, B. Ramabhadran, and S. Dharanipragada. Factor analysis invariant to linear transformations of data. In *Proceedings of the fifth international conference on spoken language processing*, pages 397–400, 1998. 60, 61

[84] A. Gunawardana and W. J. Byrne. Discriminative speaker adaptation with conditional maximum likelihood linear regression. In *Proceedings of*

the seventh European conference on speech communication and technology, pages 1203–1206, 2001. 41, 43, 52

[85] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. Hidden conditional random fields for phone classification. In *Proceedings of the ninth European conference on speech communication and technology*, pages 1117–1120, Lisbon, Portugal, 2005. 3, 102, 103, 106, 109, 110

[86] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. In N. Cristianini, editor, *Journal of machine learning research*, **46**, pages 389–422. Kluwer Academic Publishers, 2002. 78

[87] R. Haeb-Umbach. Automatic generation of phonetic regression class trees for MLLR adaptation. *IEEE Transactions on Speech and Audio Processing*, **9**:299–302, 2001. 55, 56

[88] D. Hakkani-Tur, F. Bechet, G. Riccardi, and G. Tur. Beyond ASR 1-best: Using word confusion networks in spoken language understanding. *Computer Speech and Language*, **20**[4]:495–514, 2006. 3, 33

[89] X. He and L. Deng. *Discriminative learning for speech recognition. Theory and practice.* Morgan & Claypool publishers, 2008. Synthesis lectures on speech and audio processing # 4. 2, 43

[90] G. Heigold. *A log-linear discriminative modeling framework for speech recognition.* PhD thesis, Aachen University, 2010. 71, 103, 120

[91] G. Heigold, R. Schlüter, and H. Ney. On the equivalence of Gaussian HMM and gaussian HMM-like hidden conditional random fields. In *Proceedings of the tenth European conference on speech communication and technology*, pages 1721–1724, 2007. 103, 120

[92] H. Hermansky. Perceptual Linear Predictive (PLP) analysis of speech. *Journal of the Acoustic Society of America*, **87**[4]:1738–1752, 1990. 8

[93] Y. Hifny and S. Renals. Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, **17**[2]:354–365, 2009. 104, 120, 121, 127

[94] B. Hoffmeister, R. Liang, R. Schlüter, and H. Ney. Log-linear model combination with word-dependent scaling factors. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association*, pages 248–251, Brighton, UK, 2009. 124

[95] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The annals of statistics*, **36**:1171–1220, 2008. 82

[96] J. Holmes and W. Holmes. *Speech synthesis and recognition*. Taylor & Francis, 2001. 9

[97] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13**[2]:415–425, 2002. 85, 86

[98] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. and Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, 2007. 87

[99] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing*. Prentice Hall, 2001. 9

[100] Q. Huo and Y. Hu. Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions. In *Proceedings of the tenth European conference on speech communication and technology*, pages 1042–1045, Antwerp, Belgium, 2007. 60, 61

[101] T. Jaakkola and D. Haussler. Exploiting generative models in discimінative classifiers. In D. A. Kohn, editor, *Advances in Neural Information Processing Systems*, pages 487–493. MIT Press, 1999. 83, 84, 121, 122, 123

[102] A. K. Jain, P. W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**[1]:4–37, 2000. 1

[103] E. T. Jaynes. Information theory and statistical mechanics. *Physics Reviews*, **106**[4]:620–630, 1957. 69, 70

[104] E. T. Jaynes. *Probability theory: the logic of science*. Cambridge University Press, 2003. 69

[105] F. Jelinek. A fast sequential decoding algorithm using a stack. *IBM Journal on Research and Development*, **13**[6]:675–685, 1969. 31

[106] F. Jelinek. *Statistical methods for speech recognition*. MIT Press, 1998. 1, 30

[107] H. Jiang, X. Li, and X. Liu. Large margin hidden markov models for speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, **14**[5]:1584–1595, 2006. 38

[108] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of tenth European Conference on Machine Learning*, pages 137–142, 1998. 78

[109] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999. 78, 160

[110] T. Joachims. *Learning to classify text using support vector machines: methods, theory, and algorithms*. Kluwer Academic, Boston, MA, USA, 2002. 130

[111] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural SVMs. *Journal of Machine Learning Research*, **77**:27–59, 2009. 41

[112] M. I. Jordan. Graphical models. *Statistical Science*, **19**:140–155, 2004. 72

[113] B. H. Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of markov chains. Technical report, AT&T Technical Journal, 1985. 20

[114] B.-H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, **14**[4], 1992. 37

[115] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, **33**:251–272, 1991. 11, 12

[116] J. Kaiser, B. Horvat, and Z. Kačič. A novel loss function for the overall risk criterion based discriminative training of hmm models. In *Proceedings of the sixth international conference on spoken language processing*, **2**, pages 887–890, Beijing, China, 2000. 37

[117] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero. Noise adaptive training for robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, **18**:1889–1901, 2010. 60

[118] D. Kanevsky. A generalization of the baum algorithm to functions on nonlinear manifolds. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **1**, pages 473–476, Detroit, USA, 1995. 41

[119] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **35**:400–401, 1987. 30

[120] S. S. Keerthi, S. Sundararajan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A sequential dual method for large scale multi-class linear SVMs. In *Proceedings of the fourteenth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 408–416, 2008. 86

[121] S. Khudanpur and J. Wu. Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modelling. *Computer speech and language*, **14**[4]:355–372, 2000. 130, 131

[122] D. Kim and M. J. F. Gales. Adaptive training with noisy constrained maximum likelihood linear regression for noise robust speech recognition. In *Proceedings of the tenth annual conference of the international speech communication association*, pages 2383–2386, 2009. 61

[123] D. K. Kim and M. J. F. Gales. Noisy constrained maximum-likelihood linear regression for noise-robust speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, **19**[2]:315–325, 2011. 65

[124] D. Y. Kim, N. S. Kim, and C. K. Un. Model-based approach for robust speech recognition in noisy environments with multiple noise sources. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1123–1126, Rhodes, Greece, 1997. 64

[125] D. Y. Kim, C. K. Un, and N. S. Kim. Speech recognition in noisy environments using first-order vector Taylor series. *Speech Communication*, **24**[1]:39–49, 1998. 60, 61

[126] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In F. Fogelman Souliè and J. Hèrault, editors, *Nerocomputing: algorithms, architectures and applications*, **F68**, pages 41–50. Springer, 1990. 84, 85

[127] H. K. J. Kuo and Y. Gao. Maximum entropy direct models for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, **14**[3]:873–881, 2006. 2, 3, 101, 102, 103, 107, 109

[128] M. I. Layton. *Augmented statistical models for classifying sequence data.* PhD thesis, Cambridge University, 2006. 3, 4, 71, 75, 78, 83, 84, 89, 90, 91, 92, 101, 102, 104, 107, 109, 110, 111, 112, 113, 114, 119, 122, 123, 124, 125, 126, 127, 128, 129, 133, 135, 138, 141, 164, 166, 167, 172, 190

[129] M. I. Layton and M. J. F. Gales. Maximum margin training of generative kernels. Technical Report CUED/F-INFENG/TR.484, University of Cambridge, 2004. 38

[130] M. I. Layton and M. J. F. Gales. Augmented statistical models for speech recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **I**, pages 129–132, Toulouse, France, 2006. 91

[131] L. Lee and R. C. Rose. Speaker normalization using efficient frequency warping procedures. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **1**, pages 353–356, 1996. 49, 78

[132] C. J. Leggetter. *Improved acoustic modelling for HMMs using linear transformations*. PhD thesis, Cambridge University, 1995. 55, 56

[133] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, **9**:171–185, 1995. 49, 50, 55

[134] M. Lehr and I. Shafran. Learning a discriminative weighted finite-state transducer for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, **19**[5]:1360–1367, 2011. 130

[135] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell system technical journal*, **62**[4]:1035–1074, 1983. 20

[136] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero. High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 65–70, Kyoto, Japan, 2007. 60

[137] J. Li, M. Siniscalchi, and C.-H. Lee. Approximate test risk minimization through soft margin training. In *Proceedings of IEEE international*

conference on acoustics, speech, and signal processing, **4**, pages IV–653–IV–656, 2007. 38, 39

[138] X. Li. *Regularized adaptation: theory, algorithms and applications*. PhD thesis, University of Washington, 2007. 87

[139] X. Li and J. Bilmes. Regularized adaptation of discriminative classifiers. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, Toulouse, France, 2006. 87

[140] H. Liao. *Uncertainty Decoding For Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 2007. 58, 59, 60, 62, 64, 156

[141] H. Liao and M. J. F. Gales. Adaptive training with joint uncertainty decoding for robust recognition of noisy data. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **4**, pages IV–389–IV–392, Honolulu, USA, 2007. 60

[142] L. A. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Transactions on Information Theory*, **28**:729–734, 1982. 20

[143] X. Liu and M. J. F. Gales. Automatic model complexity control using marginalized discriminative growth functions. *IEEE Transactions on Speech and Audio Processing*, **15**[4]:1414–1424, 2007. 13

[144] K. Livescu, E. Fosler-Lussier, and F. Metze. Subword modeling for automatic speech recognition. *IEEE Signal Processing Magazine. Special issue on fundamental technologies in modern speech recognition*, pages 44–57, 2012. 4, 22

[145] C. Longworth. *Kernel methods for text-independent speaker verification*. PhD thesis, Cambridge University, 2010. 83, 122, 127

[146] C. Longworth and M. J. F. Gales. Derivative and parametric kernels for speaker verification. In *Proceedings of the tenth European conference on speech communication and technology*, pages 310–313, 2007. 122

[147] J. Loof, R. Schlüter, and H. Ney. Discriminative adaptation for log-linear acoustic models. In *Proceedings of eleventh annual conference of the international speech communication association*, pages 1648–1651, 2010. 77

[148] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, **63**:561–580, 1975. 8

[149] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceeding of the sixth conference on natural language learning*, pages 49–55, 2002. 73

[150] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimisation. *Computer Speech and Language*, **14**[4]:373–400, 2000. 3, 32

[151] J. D. Markel and A. H. Jr. Gray. *Linear prediction of speech*. Springer-Verlag, New York, NY, USA, 1976. 8

[152] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis from HMMs using dynamic features. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 389–392, 1996. 94

[153] A. McCallum, D. Freitag, and F. C. N. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the seventeenth international conference on machine learning*, pages 591–598, 2000. 102, 106, 107, 109

[154] E. McDermott, T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri. Discriminative training for large-vocabulary speech recognition using minimum classification error. *IEEE Transactions on Audio, Speech, and Language Processing*, **15**[1]:203–223, 2007. 41

[155] T. P. Minka. Algorithms for maximum-likelihood logistic regression. Technical Report CMU-TR-2001-758, Carnegy Mellon University, 2003. 73

[156] M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, **7**:321–350, 2002. 34

[157] M. Mohri. Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science*, **14**:957–982, 2003. 35

[158] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, **16**:69–88, 2002. 31, 34

[159] G. L. Moore. *Adaptive statistical class-based language modelling*. PhD thesis, Cambridge University, 2001. 29, 30

[160] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen. SPEECHDAT-CAR. a large speech database for automotive environments. In *Proceedings of the second international conference on language resources & evaluation*, 2000. 156

[161] P. J. Moreno. *Speech Recognition in Noisy Environments*. PhD thesis, Carnegie Mellon University, 1996. 49, 58, 59, 60, 61, 62, 64

[162] P. J. Moreno, B. Raj, and R. Stern. A vector taylor series approach for environment-independent speech recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 733–736, Atlanta, USA, 1996. 78

[163] J. Morris and E. Fosler-Lussier. Conditional random fields for integrating local discriminative classifiers. *IEEE Transactions on Audio, Speech and Language Processing*, **16**[3]:617–628, 2008. 107, 121

[164] A. Nádas. 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, 1983. 36

[165] A. Nadás, D. Nahamoo, and M. A. Picheny. On a model robust training method for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36**:1432–1436, 1988. 35

[166] S. Narayanan and A. Alwan (editors). *Text to speech synthesis: new paradigms and advances*. Prentice Hall, 2004. 4, 92, 188

[167] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, USA, 2001. MIT Press. 69

[168] P. Nguyen, G. Heigold, and G. Zweig. Speech recognition with flat direct models. *IEEE Journal of Selected Topics in Signal Processing*, **4**:994–1006, 2010. 71, 74, 89, 101, 119

[169] P. Nguyen and G. Zweig. Extensions to the SCARF framework. Technical Report MSR-TR-2010-129, Microsoft Research, 2010. 73, 109, 112, 113

[170] K. Nigam, J. D. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999. 69, 70, 71, 73, 74

[171] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. 19, 73

[172] Y. Normandin. *Hidden Markov models, maximum mutual information estimation, and the speech recognition problem*. PhD thesis, McGill University, 1991. 21, 24, 25, 41, 43

[173] F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceeding of the fortieth annual meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, USA, 2002. 69

[174] J. J. Odell. *The use of context in large vocabulary speech recognition*. PhD thesis, Cambridge University, 1995. 22, 23, 24, 25, 26, 27, 31, 32

[175] A. V. Oppenheim. Digital processing of speech. In A. V. Oppenheim, editor, *Applications of Digital Signal Processing*, chapter 3, pages 117–168. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1978. 8

[176] A. V. Oppenheim and R. Schafer. Homomorphic analysis of speech. *IEEE Transactions on Audio and Electroacoustics*, **16**:221–226, 1968. 8

[177] S. Ortmanns, H. Ney, and X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, **11**[1]:43–72, 1997. 31

[178] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997. 78

[179] M. Padmanabhan, G. Saon, and G. Zweig. Lattice-based unsupervised MLLR for speaker adaptation. In *Proceedings of the international speech communication association tutorial and research workshop on automatic speech recognition: challenges for the new millenium*, pages 128–132, Paris, France, 2000. 52

[180] K. A. Papineni. Discriminative training via linear programming. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **2**, pages 561–564, 1999. 38

[181] N. Parihar and J. Picone. Aurora working group: DSR front end LVCSR evaluation. Technical Report AU/384/02, Mississippi State University, 2002. 157

[182] D. B. Paul and J. M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362, 1992. 156, 157

[183] D. Pearce and H.-G. Hirsch. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy

conditions. In *Proceedings of the sixth international conference on spoken language processing*, pages 29–32, Beijing, China, 2000. 91, 159

[184] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition.* PhD thesis, Cambridge University, 2003. 34, 35, 38, 41, 42, 43, 44, 45, 46, 47, 48, 112, 142, 144, 146

[185] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted mmi for model and feature-space discriminative training. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 4057–4060, 2008. 39, 40, 41

[186] D. Povey and P. C. Woodland. Large scale discriminative training of hidden Markov models in speech recognition. *Computer Speech and Language*, **16**[1]:25–48, 2002. 35, 43

[187] D. Povey and P. C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages I–105–I–108, 2002. 43, 46

[188] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, **77**, pages 257–286, February 1989. 1, 2, 9, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22

[189] L. R. Rabiner and S. E. Levinson. Isolated and connected word recognition - theory and selected applications. In A. Waibel and K.-F. Lee, editors, *Readings in speech recognition*, pages 115–153. Morgan Kaufmann Publishers, Inc., 1990. 8, 9

[190] A. Ragni and M. J. F. Gales. Derivative kernels for noise robust ASR. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 119–124, 2011. i

[191] A. Ragni and M. J. F. Gales. Structured discriminative models for noise robust continuous speech recognition. In *Proceedings of IEEE in-*

ternational conference on acoustics, speech, and signal processing, pages 4788–4791, Prague, Czech Republic, 2011. i, 4, 26

[192] A. Ragni and M. J. F. Gales. Inference algorithms for generative score-spaces. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 4149–4152, Kyoto, Japan, 2012. i

[193] J. C. Rajapakse, L. Wong, and R. Acharya (editors). *Pattern recognition in bioinformatics*, **4146**. Springer, 2006. Lecture notes in computer science. 1

[194] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, **10**:19–41, 2000. 122

[195] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *Proceedings of IEEE international conference on neural networks*, **1**, pages 586–591, 1993. 73, 168

[196] R. Rifkin and A. Klautau. In defence of one-versus-all classification. *Journal of Machine Learning Research*, **5**:101–141, 2004. 85, 86

[197] B. Roark, M. Saraclar, M. Collins, and M. Johnson. Discriminative language modelling with conditional random fields and the perceptron algorithm. In *Proceedings of the forty-second annual meeting of the association for computational linguistics*, pages 47–54, 2004. 130

[198] F. Rosenblatt. The perceptron - a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957. 40

[199] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, 1994. 30, 69, 70, 71

[200] D. B. Rubin and T. Thayer. EM algorithm for ML factor analysis. *Psychometrika*, **47**:69–76, 1982. 60, 61

[201] A. Sankar and C.-H. Lee. A maximum-likelihood approach to stochastic matching for robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, **4**:190–202, 1996. 61

[202] G. Saon, A. Dharanipragada, and D. Povey. Feature-space Gaussianization. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **1**, pages 329–332, 2004. 78

[203] G. Saon and D. Povey. Penalty function maximization for large margin hmm training. In *Proceedings of the ninth annual conference of the international speech communication association*, pages 920–923, 2008. 39, 40, 41

[204] S. Sarawagi and W. Cohen. Semi-Markov conditional random fields for information extraction. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1185–1192. MIT Press, Cambridge, MA, 2005. 108

[205] L. K. Saul and M. G. Rahim. Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, **8**:115–125, 2000. 60

[206] R. W. Schafer and L. R. Rabiner. Digital representations of speech signals. *Proceedings of the IEEE*, **63**:662–677, 1975. 9

[207] R. Schlüter, W. Macherey, B. Müller, and H. Ney. Comparison of discriminative training criteria and optimization methods for speech recognition. *Speech Communication*, **34**:287–310, 2001. 41

[208] F. Sha and L. K. Saul. Large margin gaussian mixture modelling for phonetic classification and recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 265–268, 2006. 38, 39

[209] F. Sha and L. K. Saul. Large margin hidden markov models for automatic speech recognition. In B. Schölkopf, J. Platt, and T. Hof-

MANN, editors, *Advances in Neural Information Processing Systems 19*, pages 1249–1256. MIT Press, 2007. 41

[210] I. SHAFRAN AND K. HALL. Corrective models for speech recognition of inflected languages. In *Proceedings of the conference on empirical methods in natural language processing*, pages 390–398, 2006. 131

[211] S. SHALEV-SHWARTZ, Y. SINGER, AND N. SREBRO. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proceedings of the twenty fourth international conference on Machine learning*, pages 807–814, 2007. 41

[212] C. E. SHANNON. A mathematical theory of communication. *Bell System Technical Journal*, **27**:379–423, 1948. 22, 29

[213] K. SHINODA AND T. WATANABE. Speaker adaptation with autonomous model complexity control by MDL principle. In *Proceedings of the fourth European conference on speech communication and technology*, pages 1143–1146, 1995. 56

[214] P. SHIVASWAMY AND T. JEBARA. Relative margin machines. In D. KOLLER, D. SCHUURMANS, Y. BENGIO, AND L. BOTTOU, editors, *Advances in Neural Information Processing Systems 21*, pages 1481–1488. MIT Press, 2009. 83

[215] D. A. SMITH AND J. EISNER. Minimum risk annealing for training log-linear models. In *Proceeding of the joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics*, pages 787–794, Sydney, Australia, 2006. 76

[216] N. D. SMITH. *Using Augmented Statistical Models and Score Spaces for Classification*. PhD thesis, Cambridge University, 2003. 83, 91, 121, 122, 123, 124

[217] N. D. SMITH AND M. J. F. GALES. Speech recognition using SVMs. In S. BECKER T. G. DIETTERICH AND Z. GHAHRAMANI, editors, *Advances in neural information processing systems 14*, **2**, pages 1197–1204. MIT Press, 2002. 38, 69, 83, 84, 89, 101, 119, 122, 125

[218] A. Stolcke, E. Brill, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 163–166, Rhodes, Greece, 1997. 3, 33

[219] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman. MLLR transforms as features in speaker recognition. In *Proceedings of the ninth European conference on speech communication and technology*, pages 2425–2428, 2005. 122

[220] Y.-H. Sung. *Hidden conditional random fields for speech recognition.* PhD thesis, Stanford University, 2010. 102

[221] Y.-H. Sung, C. Boulis, and D. Jurafsky. Maximum conditional likelihood linear regression and maximum a posteriori for hidden conditional random fields speaker adaptation. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 4293–4296, Las Vegas, USA, 2008. 77, 116

[222] Y.-H. Sung, C. Boulis, C. Manning, and D. Jurafsky. Regularization, adaptation, and non-independent features improve hidden conditional random fields for phone classification. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 347–352, 2007. 77, 102, 116

[223] Y.-H. Sung and D. Jurafsky. Hidden conditional random fields for phone recognition. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 107–112, 2007. 101, 102, 109

[224] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006. 2, 7, 106

[225] C. Sutton, A. McCallum, and F. C. N. Pereira. Dynamic conditional random fields: factorized probabilistic models labeling and segmenting sequence data. *Journal of Machine Learning Research*, **8**:693–723, 2007. 104

[226] P. H. Swain. Pattern recognition: a basis for remote sensing data analysis. Technical Report 111572, Purdue University, 1972. 1

[227] M. A. Tahir, G. Heigold, C. Plahl, R. Schlüter, and H. Ney. Log-linear framework for linear feature transformations in speech recognition. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 76–81, Merano, Italy, 2009. 77

[228] B. Taskar. *Learning structured prediction models: a large margin approach*. PhD thesis, Stanford University, 2004. 38, 76, 77

[229] C. W. Therrien. *Decision estimation and classification*. John Willey & Sons, 1989. 125

[230] H. Thompson. Best-first enumeration of paths through a lattice an active chart parsing solution. *Computer Speech and Language*, **4**[3]:263–274, 1990. 32

[231] T. Toda and K. Tokuda. A speech parameter generation algorithm considering global variance for hmm-based speech synthesis. *IEICE Transactions on Information and Systems*, **E90-D**:816–824, 2007. 96

[232] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 660–663, 1995. 94, 95

[233] K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, and S. Imai. An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. In *Proceedings of the fourth European conference on speech communication and technology*, pages 757–760, 1995. 2, 93, 94

[234] K. Tokuda, T. Yoshimura, T. Masuko, and T. Kobayashi, T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **3**, pages 1315–1318, 2000. 2, 93, 94, 96

[235] K. Tokuda, H. Zen, and A. W. Black. An HMM-based approach to multilingual speech synthesis. In S. Narayanan and A. Alwan, editors, *Text to speech synthesis: new paradigms and advances*, chapter 7, pages 135–153. Prentice Hall, 2004. 93, 94, 188

[236] K. Tokuda, H. Zen, and T. Kitamura. Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features. In *Proceedings of the eighth European conference on speech communication and technology*, pages 3189–3192, 2003. 97

[237] S. Tsakalidis, V. Doumpiotis, and W. J. Byrne. Discriminative linear transforms for feature normalisation and speaker adaptation in HMM estimation. *IEEE Transactions on Speech and Audio Processing*, **13**:367–376, 2005. 52

[238] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, **6**:1453–1484, 2005. 22, 41, 116

[239] L. F. Uebel and P. C. Woodland. Improvements in linear transforms based speaker adaptation. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 49–52, 2001. 52

[240] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. MMIE training of large vocabulary recognition systems. *Speech Communication*, **22**:303–314, 1997. 34, 45

[241] R. C. van Dalen, A. Ragni, and M. J. F. Gales. Efficient decoding with continuous rational kernels using the expectation semiring. Technical Report CUED/F-INFENG/TR674, Cambridge University, 2012. i

[242] R. C. van Dalen, A. Ragni, and M. J. F. Gales. Efficient decoding with generative score-spaces using the expectation semiring. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, 2013. i

[243] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995. 78, 79

[244] V. Vapnik. *Statistical learning theory*. John Wiley & Sons, 1998. 38, 69, 78, 79, 81, 82, 84, 85

[245] V. Venkataramani. *Code breaking for automatic speech recognition*. PhD thesis, Johns Hopkins University, 2005. 3, 188

[246] V. Venkataramani and W. Byrne. Lattice segmentation and support vector machines for large vocabulary continuous speech recognition. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 817–820, 2005. 89, 91, 92, 164

[247] V. Venkataramani, S. Chakrabartty, and W. J. Byrne. Support vector machines for segmental minimum Bayes risk decoding of continuous speech. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 13–18, 2003. 3, 33, 89, 90, 91

[248] V. Venkataramani, S. Chakrabartty, and W. J. Byrne. *Gini*support vector machines for segmental minimum Bayes risk decoding of continuous speech. *Computer Speech and Language*, **21**:423–442, 2007. 69, 164

[249] O. Viikki and K. Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, **25**:133–147, 1998. 78

[250] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, **4**[1]:52–57, 1968. 1

[251] A. J. Viterbi. Error bounds for convolutional codes and asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, **13**:260–269, 1982. 106, 107

[252] F. Wallhof, D. Willett, and G. Rigoll. Frame-discriminative and confidence-driven adaptation for LVCSR. In *Proceedings of IEEE interna-*

tional conference on acoustics, speech, and signal processing, pages 1835–1838, 2000. 52

[253] L. Wang and P. C. Woodland. Discriminative adaptive training using the MPE criterion. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 279–284, St. Thomas, Virgin Islands, USA, 2003. 52, 54, 55, 151

[254] S. Watanabe, T. Hori, and A. Nakamura. Large vocabulary continuous speech recognition using WFST-based linear classifier for structured data. In *Proceedings of eleventh annual conference of the international speech communication association*, pages 346–349, 2010. 130

[255] K. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of european symposium on artificial neural networks*, **4**, pages 219–224, 1999. 84, 85

[256] S. Wiesler, M. Nussbaum-Thom, G. Heigold, R. Schlüter, and H. Ney. Investigations on features for log-linear acoustic models in continuous speech recognition. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 52–57, Merano, Italy, 2009. 120

[257] S. Wiesler, A. Richard, Y. Kubo, R. Schlüter, and H. Ney. Feature selection for log-linear acoustic models. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 5324–5327, 2011. 121, 127

[258] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Yound. Large vocabulary continuous speech recognition using HTK. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **2**, pages II/125–II/128, 1994. 156

[259] P. C. Woodland, D. Pye, and M. J. F. Gales. Iterative unsupervised adaptation using maximum likelihood linear regression. In *Proceedings of the fourth international conference on spoken language processing*, pages 1133–1136, Philadelphia, PA, USA, 1996. 51

[260] Y.-J. Wu and R.-H. Wang. Minimum generation error training for HMM-based speech synthesis. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 89–92, 2006. 96, 97

[261] H. Xu, M. J. F. Gales, and K. K. Chin. Improving joint uncertainty decoding performance by predictive methods for noise robust speech recognition. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 222–227, 2009. 63, 64

[262] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modelling for HMM-based speech synthesis. In *Proceedings of the fifth international conference on spoken language processing*, **2**, pages 29–32, 1998. 93

[263] S. Young. Statistical modelling in continuous speech recognition. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 562–571, 2001. 1, 2, 8

[264] S. J. Young. A review of large-vocabulary continuous-speech recognition. *IEEE Signal Processing Magazine*, **13**:45–57, 1996. 22

[265] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book (for HTK Version 3.4.1)*. University of Cambridge, `http://htk.eng.cam.ac.uk`, May 2009. 3, 8, 10, 13, 15, 16, 17, 18, 19, 21, 22, 26, 29, 30, 31, 32, 33, 34, 35, 44, 45, 48, 49, 50, 52, 56, 144, 160, 167, 169, 170, 178, 179

[266] S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of ARPA Workshop on Human Language Technology*, pages 307–312, 1994. 24, 25, 26, 27, 28, 138

[267] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proceedings of the 26-th international conference on machine learning*, pages 1169–1176, Montreal, Canada, 2009. 116

[268] K. Yu, H. Zen, F. Mairesse, and S. Young. Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis. *Speech communication*, **53**:914–923, 2011. 26

[269] H. Zen. *Reformulating HMM as a trajectory model by imposing explicit relationships between static and dynamic features*. PhD thesis, Nagoya Institute of Technology, 2006. 93, 94, 95, 96, 97

[270] H. Zen, Y. Nankaku, and K. Tokuda. Model-space MLLR for trajectory HMMs. In *Proceedings of the tenth European conference on speech communication and technology*, pages 294–299, 2007. 2, 188

[271] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda. The HMM-based speech synthesis system (HTS) version 2.0. In *Proceedings of the sixth international speech communication association workshop on speech synthesis*, pages 294–299, Bonn, Germany, 2007. 160

[272] H. Zen, K. Tokuda, and A. W. Black. Statistical parametric speech synthesis. *Speech Communication*, **51**:1039–1064, 2009. 93, 188

[273] H. Zen, K. Tokuda, and T. Kitamura. A Viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, **1**, pages I–837–I–840, 2004. 97

[274] H. Zen, K. Tokuda, and T. Kitamura. Reformulating the hmm as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech and Language*, **21**:153–173, 2007. 94

[275] S.-X. Zhang and M. J. F. Gales. Extending noise robust structured support vector machines to larger vocabulary tasks. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 18–23, Big Island, Hawaii, USA, 2011. 76, 77, 116, 124

[276] S.-X. Zhang and M. J. F. Gales. Structured support vector machines for noise robust continuous speech recognition. In *Proceedings of twelfth annual conference of the international speech communication association*, pages 989–992, 2011. 107, 108, 115

[277] S.-X. Zhang and M. J. F. Gales. Structured SVMs for automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, (*to appear*), 2012. 41, 86, 108, 115, 116

[278] S.-X. Zhang, A. Ragni, and M. J. F. Gales. Structured log-linear models for noise robust speech recognition. *IEEE Signal processing letters*, **17**:945–948, 2010. i, 3, 4, 76, 77, 85, 86, 91, 110, 113, 115, 116, 123, 124, 129, 133, 135, 138, 141, 161, 166, 167, 168, 172, 190

[279] J. Zheng and A. Stolcke. Improved discriminative training using phone lattices. In *Proceedings of the ninth European conference on speech communication and technology*, pages 2125–2128, Lisbon, Portugal, 2005. 38, 39

[280] G. Zweig and S. Chang. A comparison of algorithms for maximum entropy parameter estimation. In *Proceeding of the twelfth annual conference of the international speech communication association*, pages 609–612, 2011. 73

[281] G. Zweig and P. Nguyen. A segmental CRF approach to large vocabulary continuous speech recognition. In *Proceedings of IEEE workshop on automatic speech recognition and understanding*, pages 152–157, 2009. 3, 101, 102, 104, 106, 107, 108, 109, 110, 111, 119, 121, 130, 133

[282] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, F. Sha, M. Wang, A. Jansen, H. Hermansky, K. Karakos, D. Kintzley, S. Thomas, G. S. V. S Sivaram, S. Bowman, and J. Kao. Speech recognition with segmental conditional random fields: final report from the 2010 JHU summer workshop. Technical Report MSR-TR-2010-173, Microsoft Reasearch, 2010. 112, 113

[283] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, G. S. V. S Sivaram, S. Bowman, and J. Kao. Speech recognition with segmental conditional random fields: a summary of the JHU CLSP 2010 summer workshop. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing*, pages 5044–5047, Prague, Czech Republic, 2011. 121, 133