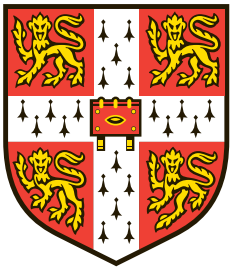
Structured and Infinite Discriminative Models for Speech Recognition

Jingzhou Yang
Homerton College



Department of Engineering
University of Cambridge

A thesis submitted to the University of Cambridge for the degree of
Doctor of Philosophy

December 2015

To my family.

Structured and Infinite Discriminative Models for Speech Recognition

Jingzhou Yang

A thesis submitted to the University of Cambridge for the degree of Doctor of Philosophy

Submitted Dec. 2015 Revised Jul. 2016

Abstract

In continuous speech recognition, observations are sequential data with variable length, and labels are sequence of words (or sub-words) possibly having unbounded number of classes. It is thus impractical to robustly construct models for the whole word sequence. To address this problem, rather than treating the whole sentence as an atomic unit, structure needs to be introduced into classifiers to break the sentence label into words or sub-word units. These are usually referred to as structured discriminative models, where the conditional distribution of the classes given the observations is directly modelled. Compared with generative models, discriminative models have the potential to improve performance as a wide range of features from the observation and word sequences can be used. Moreover, in application of generative models, such as the hidden Markov model (HMM), the frame-level Markov assumption is often assumed. However, discriminative models are much easier to deal with segment level modelling, where the frame level Markov assumption is relaxed. Then long-span dependencies among observations are allowed to be captured.

One major contribution of this thesis is the study of the features based on generative models. Since speech observations are sequential data with variable length, it is not obvious how to extract features from these sequences. Sequential kernels are often used to map observations into a space with fixed dimensions. In this work, generative models are simply used to provide such mapping, where the extracted features have a compact form. Moreover, the baseline performance of generative models can be retrieved by discriminative models, e.g. when using the features comprised of the likelihoods from HMMs. In the past few years, deep neural networks (DNNs) have been widely used in speech community, and significant performance gains have been achieved. This thesis thus focuses on the features extracted from the DNN-based systems, such as hybrid and tandem. In order to make use of the complementary information from different systems, features based on multiple systems are also studied.

The commonly used discriminative models, such as log-linear models, only yield linear decision boundaries. One solution to this problem is the use of the "kernel trick". Alternatively, the mixture-of-experts framework can be employed. In this framework, multiple experts are used in classification, that allows an overall non-linear decision boundary. However, it might be problematic to choose the number of experts. In order to sidestep of the problem of setting the model complexity, the Bayesian non-parametric framework can be used. In Bayesian non-parametric approaches, rather than specifying the model complexity in advance, the model complexity is part of the posterior inference. When making predic-

tions, the posterior distribution of the model parameters can be integrated over, effectively averaging over models of all possible complexity.

Another major contribution of this thesis is the extension of the structured discriminative models to Bayesian non-parametrics. The infinite structured support vector machine (SVM) is one such example. It is also a structured extension of the infinite SVM, which is a mixture-of-experts model using SVMs as experts. Bayesian inference can be viewed as a particular minimisation criterion. This thesis extends this specific criterion to a more general form. Then alternative criteria can be derived, such as the large margin training criterion which has good generalisation properties.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This thesis contains around 61,000 words including appendices, bibliography, footnotes, tables and equations. It has 47 figures and 16 tables.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Prof. Mark Gales, for his guidance and consistent help for these years. I would also like to express my deep appreciation to the EPSRC and Cambridge Overseas Trust for funding my PhD study.

I owe a debt of gratitude to my colleagues in the Machine Intelligence Lab for their help and encouragements given to me. Particular thanks must go to Rogier van Dalen, Anton Ragni, Kate Knill, Shi-Xiong Zhang, Yongqiang Wang, Xunying Liu, Kai Yu, Yanhua Long, Xie Chen, Chao Zhang, Chunyang Wu, Linlin Wang, Haipeng Wang, Yanming Qian and Yu Wang who make the study in this lab enjoyable and fruitful. Special thanks to Patrick Gosling and Anna Langley for their excellent work in creating and maintaining the outstanding computing facilities. Next to my examiners, Dr. Richard Turner and Prof. Mahesan Niranjan, who gave many valuable comments on this thesis.

Finally, I would like to thank my parents for all their support over the years.

Contents

List of Figures

List of Tables

Acronyms

ASR	Automatic Speech Recognition
CAug	Conditional Augmented Model
CCCP	Concave-Convex Procedure
CML	Conditional Maximum Likelihood
CMLLR	Constrained Maximum Likelihood Linear Regression
CRFs	Conditional Random Fields
CRP	Chinese Restaurant Process
CSR	Continuous Speech Recognition
DCT	Discrete Cosine Transform
DNN	Deep Neural Network
DP	Dirichlet Process
EM	Expectation Maximisation
FLP	Full Language Pack
GMM	Gaussian Mixture Model
HCRFs	Hidden Conditional Random Fields
HDP	Hierarchical Dirichlet Process
HMI	Human-Machine Interaction
HMM	Hidden Markov Model
iGMM	infinite Gaussian Mixture Model
iHMM	infinite Hidden Markov Model

iLLM	infinite Log-Linear Model
iSDM	infinite Structured Discriminative Model
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
LLM	Log-Linear Model
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MBR	Minimum Bayes Risk
MC	Monte Carlo
MCE	Minimum Classification Error
MCMC	Markov Chain Monte Carlo
MEMM	Maximum Entropy Markov Model
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information
MPE	Minimum Phone Error
MPFE	Minimum Phone Frame Error
MWE	Minimum Word Error
NLP	Natural Language Processing
PMC	Parallel Model Combination
SCRFs	Segmental Conditional Random Fields
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis
VLLP	Very Limited Language Pack
VTS	Vector Taylor Series

WER	Word Error Rate
WSJ	Wall Street Journal

Chapter 1

Introduction

Language is the system of communication used by humans [25, 134], and speech is the verbal means of communicating [4], which is the language used when speaking [25, 134]. The origins of language and speech are unknown and subject to much debate and speculation. Humans have speculated about the origins of language throughout history. The Biblical story of the Tower of Babel [129] is one such account; other cultures have different stories of how language arose [82]. Given the diversity of languages and interest of human-machine interaction (HMI) [26], speech recognition is a compelling technology that makes efficient communication between humans (speaking different languages) and machines possible [47]. The majority of automatic speech recognition (ASR) systems use hidden Markov models (HMMs) as the underlying acoustic models, and significant improvement can be achieved by employing discriminative training [7, 100, 130, 140]. Moreover, speech recognition is a classification task. It would, therefore, be interesting to examine discriminative models for speech recognition [38, 54, 65, 109, 147, 209, 215], where the conditional distribution of the word sequence given the observations is directly modelled. In this thesis, a class of discriminative models called *structured discriminative models* will be studied.

In machine learning, one of the main issues that might be encountered is the mismatch between model complexity and the amount of training data available [133, 177]. The traditional parametric model with fixed and finite number of parameters might suffer from the problem of over-fitting or under-fitting [177]. Thus a Bayesian non-parametric model might

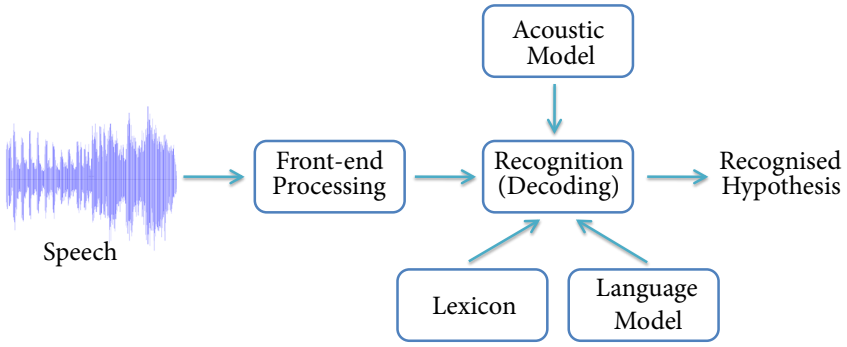


Figure 1.1: *A typical automatic speech recognition system.*

be a better choice, where the problem of choosing model complexity can be sidestepped [133, 149]. Moreover, as a Bayesian model, the problem of over-fitting can be mitigated [177]. This motivates the application of Bayesian non-parametric models in speech recognition [196, 197], which will also be studied in this thesis. Before the discussion of discriminative models and Bayesian non-parametric models for speech recognition, basic speech recognition systems and how the discriminative models and non-parametric models being applied in speech recognition will be briefly introduced in the following sections.

1.1 Speech Recognition Systems

The state-of-the-art speech recognition systems are based on statistical approaches, and a speech recognition system is normally decomposed into individual parts. The structure of a typical ASR system is illustrated in Figure 1.1. Given the speech inputs, a sequence of words associated with the inputs can be recognised through the recognition system. As illustrated in Figure 1.1, at the first stage of speech recognition, through front-end processing (or feature extraction), the speech signal is compressed into a sequence of feature vectors which are also referred to as observations, denoted by $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$. Given the observations, acoustic model, lexicon and language model, the hypothesised word sequence can be generated, which is denoted by W. In large vocabulary continuous speech recognition (LVCSR), the lexicon (also referred to as the dictionary) dictates how the sub-word units (from which the acoustic models are constructed) are linked together to form individual words. For example, the word thesis can be broken up as [30]:

$$\text{thesis} = \{\text{th iy s ih s}\}.$$

The language model contains the information about which word sequences are allowable, and it gives a probability distribution over these word sequences. The acoustic model represents the relationship between observations and the sub-word units that make up speech.

For the speech recognition systems based on statistical approaches, given the observations $\boldsymbol{O}$, the most likely word sequence $\hat{W}$ can be obtained by using the Bayesian decision rule [16]:

$$\hat{W} = \arg\max_{W} P(W|\boldsymbol{O}) \tag{1.1}$$

By using Bayes' rule, the decision rule described in (1.1) can be further written as follows:

$$\begin{aligned} \hat{W} &= \arg\max_{W} \frac{p(\boldsymbol{O}|W)P(W)}{p(\boldsymbol{O})} \\ &= \arg\max_{W} p(\boldsymbol{O}|W)P(W) \end{aligned} \tag{1.2}$$

where $p(\boldsymbol{O})$ is not a function of the word sequence W, hence it can be omitted in the decision rule. The distribution $p(\boldsymbol{O}|W)$ is given by the acoustic model. The majority of speech recognition systems are based on the HMM acoustic model or its variants, e.g. the hybrid system where the state likelihoods are given by deep neural networks (DNNs) [85] rather than Gaussian mixture models (GMMs). $P(W)$ is the probability of the word sequence W given by the language model. If the word sequence W is comprised of $\{w_1, \cdots, w_I\}$, the probability $P(W)$ can be described as:

$$P(W) = \prod_{i=1}^{I} P(w_i|w_1, \cdots, w_{i-1}) \tag{1.3}$$

Due to the vocabulary size being very large in LVCSR, it is infeasible to estimate $P(W)$ for every possible word sequence robustly. Often the N-gram language model is used, in which the probability of the current word is assumed to be only dependent on the previous $N-1$ words. Then the probability $P(W)$ in (1.3) can be further written as:

$$P(W) = \prod_{i=1}^{I} P(w_i|w_{i-N+1}, \cdots, w_{i-1}) \tag{1.4}$$

The most commonly used values for N are 1, 2 and 3. These settings are called unigram, bigram and trigram language models respectively. Due to data sparsity, normally the smoothing schemes such as discounting, back-off and deleted interpolation are used [101].

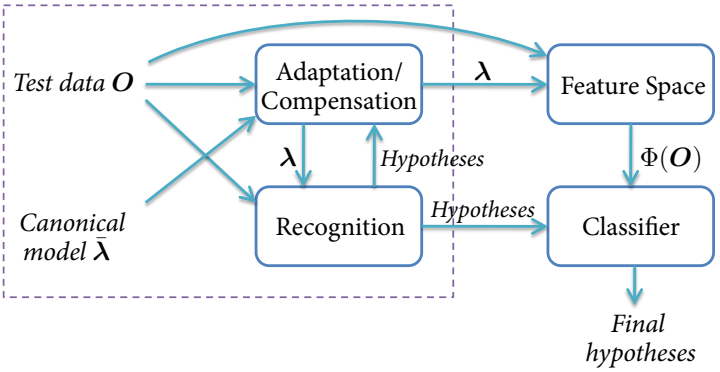


Figure 1.2: *The diagram of classifiers based on feature spaces. The region associated with the dotted rectangle illustrates a state-of-the-art speech recognition system.*

1.2 Discriminative Models and Bayesian Non-parametric Models in Speech Recognition

Speech recognition is the problem of classifying the sequential data, and the input sequences have various length. Normally, the classifiers, such as support vector machines (SVMs) [183], cannot directly deal with the features with various length. One solution to this problem is to use the feature space (or score space) [148] based on generative models, which maps the input sequences with various lengths to a space with a fixed dimension, e.g. the log-likelihood feature space is commonly used [37, 60, 148, 210]. By using the feature space, discriminative and Bayesian non-parametric models (or classifiers) then can be applied in speech recognition. The diagram of classifiers based on feature spaces is illustrated in Figure 1.2. In this figure, the region associated with the dotted rectangle is a state-of-the-art speech recogniser. The standard adaptation and model compensation schemes can be employed by the recogniser to generate the compensated feature vectors. Thus, one advantage of using this framework is that the features extracted from the compensated models (HMMs) can be automatically adapted to the target noise/speaker conditions [148]. Then, at the final stage, classifiers can be trained with these robust features. Another main advantage is the nature of the feature space based on generative models. Generative models such as HMMs have underlying conditional independence assumption that, whilst enabling them to efficiently represent data sequences, does not accurately represent the dependencies in data sequences such as speech. The feature space associated with a generative model

does not have the same conditional independence assumption as the original generative model. This allows more accurate modelling of the dependencies in the speech data [37].

By using the framework illustrated in Figure 1.2, the classifiers can be trained based on the feature space generated by the speech recognition system using deep neural networks (DNNs) [85], e.g. the tandem system [84] where the outputs from a DNN are appended to each feature vector. In additional to a single system, multiple systems can also be used in generating the feature vectors for classifiers [38], which will be discussed in detail in Chapter 3.

1.3 Thesis Organisation

This thesis can be split into 3 main parts. Parametric models will be discussed in the first part which starts from Chapter 2 to 3. In the first part, the commonly used generative and discriminative models for speech recognition will be introduced. Non-parametric models will be discussed in the second part, which starts from Chapter 4 to 5. In the second part, some of the commonly used Bayesian non-parametric models will be introduced, and the infinite structured discriminative model will be studied. Finally, in the third part (Chapter 6 and 7) the experimental results and conclusions will be discussed. The supplementary knowledge and further discussions will also be given in the appendices.

One major contribution of this thesis is the study of the features (for structured discriminative models) extracted from the DNN-based systems, such as hybrid and tandem. In order to make use of the complementary information from different systems, features based on multiple systems are also studied in this thesis. These are discussed in Chapter 3. Another major contribution of this thesis is the extension of the structured discriminative models to Bayesian non-parametrics, which are discussed in Chapter 5, as well as in Appendix E and F which give more details. This thesis has 7 chapters, and a brief chapter-by-chapter breakdown is given as follows.

Chapter 2 The widely used generative models in speech recognition, such as the hidden Markov model (HMM), and various commonly used training criteria in speech recognition will be introduced. Adaptation and noise robustness will also be briefly discussed.

Chapter 3 The extensively used unstructured and structured discriminative models in speech recognition, and various training criteria will be introduced. Different forms of the features for discriminative models will also be discussed in this chapter.

Chapter 4 The motivation of research on Bayesian non-parametric models will be presented, and some of the commonly used Bayesian non-parametric models will be discussed.

Chapter 5 A criterion-based perspective on Bayesian inference will be introduced. Bayesian inference and large margin training of the infinite structured discriminative model will be discussed in detail.

Chapter 6 Different types of data sets, i.e. AURORA2, AURORA 4 and BABEL, and the corresponding experimental results on these sets will be represented in this chapter.

Chapter 7 Conclusions and possible directions for the future work will be discussed.

Chapter 2

Generative Models

Generative models are the most extensively used forms of statistical models in speech recognition. As discussed in Chapter 1, in speech recognition generative approaches are based on the combination of the acoustic and language models, where the posterior distribution of the word sequence $W = \{w_1, \ldots, w_I\}$ given the observations $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ can be expressed as the following form (according to Bayes' rule):

$$P(W|\boldsymbol{O}) = \frac{p(\boldsymbol{O}|W)P(W)}{p(\boldsymbol{O})} \tag{2.1}$$

where the probability $p(\boldsymbol{O})$ is class-independent, $p(\boldsymbol{O}|W)$ is the acoustic model and $P(W)$ is the language model. The most likely word sequence $\hat{W}$ can be yielded by maximising the class posterior distribution (2.1). In speech recognition hidden Markov models (HMMs) are the most popular and successful statistical acoustic models, which will be introduced in the following sections. Moreover, the commonly used generative models in speech recognition and various training criteria will be discussed.

2.1 Gaussian Mixture Models

The Gaussian distribution is one of most commonly used distribution, since it has a variety of properties, e.g. the *central limit theorem* [13] states that the average of the independent and identically distributed (i.i.d.) random variables converges to a Gaussian distribution

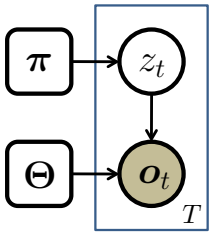


Figure 2.1: *The graphical model of a Gaussian mixture model. In the graphical model, the plate represents replication. The circle denotes variable, and the gray one denotes observation. The square represents fixed parameters. This type of representation of graphical models is used throughout this thesis.*

as the number of variables goes infinite. However, the Gaussian distribution suffers from significant limitations when it comes to modelling real data sets [16]. Then, a mixture of Gaussian distribution becomes a better choice, and this type of model is referred to as the *Gaussian mixture model* (GMM). By using a sufficient number of Gaussians, any continuous distribution can be approximated with arbitrary accuracy [16, 89]. Thus, in speech recognition the GMM is widely used in modelling the density of observations. Given the number of components M, the probability density function of the GMM can be expressed as follows [16]:

$$p(\boldsymbol{o}|\boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\boldsymbol{o}; \boldsymbol{\theta}_m) \tag{2.2}$$

where $\boldsymbol{o}$ is the observation variable, and $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_M\}$ are the *mixture weights* or *mixture coefficients*, that satisfy $\sum_m \pi_m = 1$ [16]. $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are the parameters of the Gaussian components, and for the mth component, the parameters $\boldsymbol{\theta}_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ are the mean and covariance of the Gaussian distribution.

By introducing the indicator variable z_t (that denotes which component the tth observation $\boldsymbol{o}_t$ is associated with), the generative process of the GMM can be described as follows:

$$z_t \sim \text{Categorical}(\boldsymbol{\pi}) \tag{2.3}$$

$$\boldsymbol{o}_t \sim \mathcal{N}(\boldsymbol{o}; \boldsymbol{\theta}_{z_t}) \tag{2.4}$$

where Categorical$(\cdot)$ is the *categorical distribution*, which is the generalisation of the Bernoulli distribution with multiple possible outcomes. The corresponding graphical model of the GMM is illustrated in Figure 2.1.

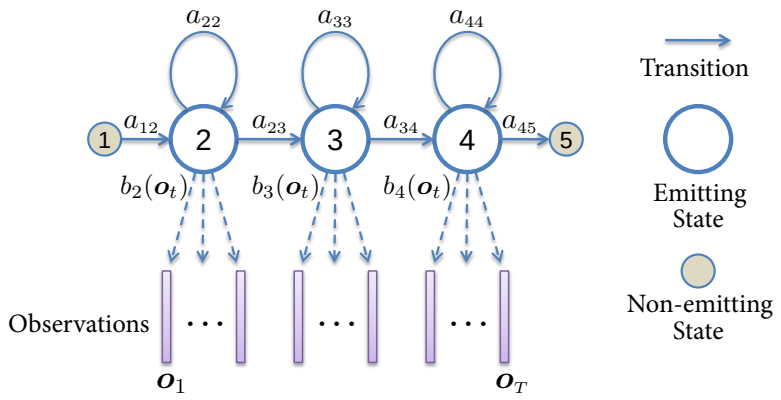


Figure 2.2: *The topology of a hidden Markov model with 3 emitting states.*

2.2 Hidden Markov Models

The Gaussian mixture model (GMM) was discussed in the previous section. Although real data sets can be well modelled by the GMM, these data are assumed to be independent and identically distributed (i.i.d.). In speech recognition the inputs are sequential data where the sequential aspects must be considered such as correlations between observations. This leads to the consideration of the hidden Markov model (HMM) [146], which is the most popular and successful statistical model in speech recognition, given its ability to modelling sequential data. As described in equation (1.2), the most likely word sequence given the observations can be determined by the HMM (acoustic model) in conjunction with the language model. The HMM is a natural extension of the Markov chain (where outputs of the state are deterministic) by using a probabilistic function associated with each state [89]. Figure 2.2 illustrates a HMM with 3 emitting states. In this figure, states 2, 3 and 4 are emitting states, where observations are generated by these states; states 1 and 5 are non-emitting states. As illustrated in Figure 2.2, the observations are $\boldsymbol{O} = \{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_T\}$, and the corresponding state sequence associated with these observations can be described as $S = \{s_1, \cdots, s_T\}$, where these states might have repeated values. Assume the number of unique states is L, then each s_t denotes one of the L states. The transition probability from state i to state j is denoted as a_{ij}, and the probability (or distribution) of an observation $\boldsymbol{o}_t$ generated by state j is described as $b_j(\boldsymbol{o}_t)$. The parameter set of the HMM is denoted as $\boldsymbol{\lambda} = \{\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{B}\}$, and the definitions of these parameters are given as follows:

- $\boldsymbol{c}$ – **Initial state distribution**

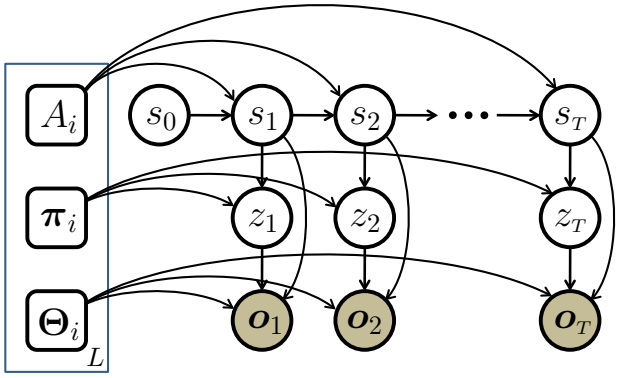


Figure 2.3: *The graphical model of the HMM with GMM state output distributions. In this figure, A_i denotes the ith row of the transition matrix $\boldsymbol{A}$. $\boldsymbol{\pi}_i$ and $\boldsymbol{\Theta}_i$ are the mixture weights and component parameters of the GMM associated with state i. z_t is the indicator variable that denotes which Gaussian component the tth observation $\boldsymbol{o}_t$ is associated with.*

The initial state distribution (of state i) can be described as:

$$c_i = P(s_0 = i), \quad \text{where } 1 \leq i \leq L \tag{2.5}$$

Since $\boldsymbol{c} = \{c_1, \ldots, c_L\}$ is a distribution, the following property must be satisfied:

$$\sum_{i=1}^{L} c_i = 1 \quad \text{and} \quad c_i \geq 0 \tag{2.6}$$

where L is the number of the unique states. As illustrated in Figure (2.2), the non-emitting states are introduced, hence the probability of the initial state that denotes state 1 is always 1, namely $P(s_0 = 1) = 1$.

- ***A* – State transition probability matrix**

  Each element of the state transition probability matrix $\boldsymbol{A}$ is defined as:

$$a_{ij} = P(s_{t+1} = j | s_t = i), \quad \text{where } \sum_{j=1}^{L} a_{ij} = 1 \text{ and } a_{ij} \geq 0 \tag{2.7}$$

  where a_{ij} is the probability of taking a transition from state i to state j. In speech recognition, since the HMMs are normally constrained to be left-to-right, the transition matrix is not necessarily full.

- ***B* – State output probability**

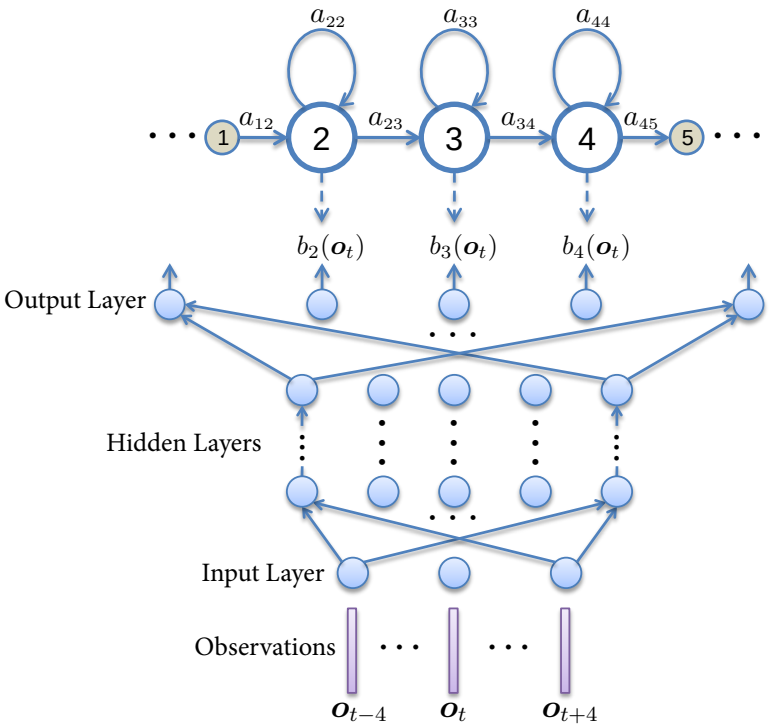


Figure 2.4: *The topology of the DNN-HMM.*

Each emitting state j is associated with one output probability distribution $b_j(\cdot)$. Given the observation $\boldsymbol{o}_t$ and the corresponding state $s_t = j$, the output distribution can be described as:

$$b_j(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|s_t = j) \tag{2.8}$$

Two forms of the state output distribution are usually adopted in state-of-the-art speech recognition systems. One form of the distribution is given by the GMM defined in equation (2.2), and this type of model is usually referred to as the GMM-HMM. Figure 2.3 illustrates the graphical model of the GMM-HMM. Alternatively, the state output distribution $b_j(\boldsymbol{o}_t)$ can be the likelihood derived from the *deep neural networks* (DNNs) [85], and the resulting framework is knows as the DNN-HMM hybrid system [23, 158]. The topology of the DNN-HMM hybrid system is illustrated in Figure 2.4. In the hybrid system, the DNN aims to model the posterior distribution of each state $P(s_t = j|\boldsymbol{o}_t)$ directly. Given the state posterior probabilities, the state output distribution for each state can be obtained by applying Bayes' rule:

$$b_j(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|s_t = j) = \frac{P(s_t = j|\boldsymbol{o}_t)p(\boldsymbol{o}_t)}{P(s_t = j)} \tag{2.9}$$

where $P(s_t = j|\boldsymbol{o}_t)$ is the state posterior probability estimated from the DNN. $P(s_t = j)$ is the prior probability of state j, which can be estimated from the training set. It is worth noting that $p(\boldsymbol{o}_t)$ can be omitted as it does not depend on a particular state.

In speech recognition, system combination approaches are of growing interest, and one of the recently proposed approach is joint decoding [191], in which the systems to be combined share the same HMM topology, and the frame level acoustic likelihoods from different systems are combined. In other words, the acoustic models of the joint decoding system can be viewed as HMMs where the state output score[1] is an combination of the state output probabilities from different HMMs (with the same topology). Normally, the logarithms of these probabilities are used. Assume there are D different systems to be combined, then the state output log score for the joint decoding system can be described as:

$$\log\left(b_j(\boldsymbol{o}_t)\right) = \sum_{d=1}^{D} \eta_d \log\left(p_d(\boldsymbol{o}_t|s_t = j)\right) \tag{2.10}$$

where the scalar η_d is the corresponding combination weight, and $p_d(\boldsymbol{o}_t|s_t = j)$ is the sate output probability given by the dth system. Normally, these sate output probabilities can be given by the GMM or DNN as described in (2.8) and (2.9). In work [191] combination of two forms of DNN based systems were investigated, i.e. the tandem (where the inputs of the GMM-HMM is appended with features from DNN) and hybrid systems [84, 85], and the combination weights used in [191] are set empirically.

The basic theory for HMMs has been presented in the previous paragraphs. In speech recognition, the application of HMMs is dependent on two assumptions [57, 201]:

- **Quasi-stationary**: Speech signals may be split into segments corresponding to states, in which the speech waveform may be considered to be stationary. The transition between these states is assumed to be instantaneous.
- **Conditional independence**: The probability of a certain observation being generated is only dependent on the current state; and given the associated state, the observation is conditional independent of both the preceding and following observations.

[1] In joint decoding system, the state output is not a valid probability, so word "score" is used instead of "probability".

Although neither of these assumptions is strictly true for speech, nevertheless HMMs are widely used in speech recognition, and state-of-the-art recognition performance can be achieved.

In order to apply HMMs to real-world implementation, three basic problems of interest need to be solved [89, 146]:

- **The evaluation problem**: Given the model $\boldsymbol{\lambda} = \{\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{B}\}$, the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ and the corresponding hypothesis (word sequence) W, how to efficiently compute the probability $p(\boldsymbol{O}|W;\boldsymbol{\lambda})$? This evaluation problem also can be viewed as the problem of evaluating how well the given model matches the given observation sequence [146]. In practice $p(\boldsymbol{O}|W;\boldsymbol{\lambda})$ can be efficiently calculated through the forward algorithm [8, 9].
- **The decoding problem**: Given the model $\boldsymbol{\lambda}$ and the observation sequence $\boldsymbol{O}$, how to choose the corresponding optimal state sequence $S = \{s_1, \ldots, s_T\}$? This is a decoding problem, where the hidden state sequence is attempted to be uncovered. In practice, an optimisation criterion is used to solve this problem, and the most widely used criterion is to find the single best state sequence. A dynamic programming approach called the *Viterbi algorithm* [46, 187] is usually employed to find this single best state sequence.
- **The training problem**: Given a set of observation sequences, how to estimate the model parameters $\boldsymbol{\lambda}$? In this training problem, normally the model parameters are attempted to be optimised so as to best describe how observations come about [146]. In practice, various training criteria can be applied to optimise the model parameters $\boldsymbol{\lambda}$, and the most commonly used optimisation criteria in speech recognition will be discussed in the following section.

2.3 Training Criteria for Generative Models

In the previous sections, the most commonly used generative models in speech recognition were discussed. The implementation of these generative models requires estimation of the model parameters. In this section, various training criteria which are used in parameter estimation will be introduced, and the hidden Markov model (HMM) will be taken as an example of the generative model in discussing these criteria. In this thesis only *supervised*

training is considered, where the training set $\mathcal{D}$ consists of utterance and reference (word or sub-word sequence) pairs:

$$\mathcal{D} = \Big\{ (\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N) \Big\} \tag{2.11}$$

In general, given the training data $\mathcal{D}$, the model parameters can be estimated by maximising (or minimising) the objective function $\mathcal{F}(\boldsymbol{\lambda})$:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \big\{ \mathcal{F}(\boldsymbol{\lambda}) \big\} \tag{2.12}$$

The most common approach for parameter estimation is *maximum likelihood* (ML) estimation [144, 156], where the model parameters are optimised by maximising the likelihood function given the training data. Alternative to ML estimation, various discriminative training criteria have been proposed. These criteria aim to optimise the model parameters according to the objective functions which are directly related to the classification performance. In the following subsections, the commonly used training criteria in speech recognition will be discussed.

2.3.1 Maximum Likelihood (ML)

In *maximum likelihood* (ML) estimation, the model parameters are optimised by maximising the probability of the observations given the word sequences and model parameters. The ML training criterion can be described as maximising the following objective function:

$$\mathcal{F}_{\text{ML}}(\boldsymbol{\lambda}) = \prod_{n=1}^{N} p(\boldsymbol{O}_n | W_n; \boldsymbol{\lambda}) \tag{2.13}$$

or

$$\mathcal{F}_{\text{ML}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \log p(\boldsymbol{O}_n | W_n; \boldsymbol{\lambda}) \tag{2.14}$$

Maximisation of these two expressions are equivalent, and they will be used interchangeably in this thesis. For the HMM, there is no analytically tractable solution that maximises the objective function (2.14) [146]. Although there is no optimal way of estimating the model parameters, a local optimal solution can be found by using the iterative procedure such as the *expectation maximisation* (EM) algorithm [39], i.e. the *Baum-Welch* algorithm [8, 9]. Alternative to ML estimation, model parameters can be estimated by using the *maximum a*

posteriori (MAP) criterion, which can be viewed as a regularization of ML estimation with a prior distribution. In speech recognition, since MAP estimation is often used in adaptation, this criterion will be introduced in the section of discussing adaption.

2.3.2 Maximum Mutual Information (MMI)

In ML estimation, training data sufficiency and model-correctness are required. However, in general neither of these requirements is strictly satisfied when modelling speech data. Thus, ML estimation may not yield the most appropriate model parameters for speech recognition. Then discriminative training approaches might be preferred, where the model parameters are optimised according to the objective functions which are directly related to the classification performance. In speech recognition one of the commonly used discriminative training criteria is the *maximum mutual information* (MMI) criterion [7, 130], where the mutual information between the utterance $\boldsymbol{O}$ and word sequence W is aimed to be maximised. Since the joint distribution $p(\boldsymbol{O}, W)$ is unknown, the empirical distribution formulated by the training samples $\{\boldsymbol{O}_n, W_n\}$ is used as an approximation. Then, the mutual information can be described as [66, 89]:

$$\mathcal{I}(\boldsymbol{O}, W; \boldsymbol{\lambda}) \approx \frac{1}{N} \sum_{n=1}^{N} \log \left(\frac{p(\boldsymbol{O}_n, W_n; \boldsymbol{\lambda})}{p(\boldsymbol{O}_n; \boldsymbol{\lambda}) P(W_n)} \right) \tag{2.15}$$

As only the acoustic model parameters are trained, $P(W_n)$ is fixed. Thus, maximising the mutual information described in (2.15) is equivalent to maximise the following objective function:

$$\mathcal{F}_{\text{MMI}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \log P(W_n | \boldsymbol{O}_n; \boldsymbol{\lambda}) \tag{2.16}$$

By using Bayes' rule, the conditional distribution $P(W_n|\boldsymbol{O}_n, \boldsymbol{\lambda})$ in (2.16) can be written in a form consisting of generative model likelihoods $p(\boldsymbol{O}_n|W_n; \boldsymbol{\lambda})$:

$$P(W_n | \boldsymbol{O}_n; \boldsymbol{\lambda}) = \frac{p(\boldsymbol{O}_n | W_n; \boldsymbol{\lambda}) P(W_n)}{\sum_W p(\boldsymbol{O}_n | W; \boldsymbol{\lambda}) P(W)} \tag{2.17}$$

In the denominator term of the right hand side of (2.17), the sum is taken over all possible hypotheses (word sequences) W including both the correct and competing ones. The number of all possible hypotheses for an utterance is exponentially large, but a N-Best list [29] or a lattice [141, 157] can be used to limit the search space of hypotheses. Since lattices

are more compact representations, they are widely used in discriminative training. If the assumption of the underlying distribution is correct and there are sufficient training data, the optimal distributions from ML estimation (discussed in section 2.3.1) and MMI estimation converge to the true underlying distribution [89]. Alternative to MMI estimation, in speech recognition other discriminative training criteria such as minimum classification error (MCE) [100] and minimum Bayesian risk (MBR) [140] are also widely used, which will be discussed in the following subsections.

2.3.3 Minimum Classification Error (MCE)

In *minimum classification error* (MCE) training, the classification error rate is to be minimised, and the goal of training is to be able to correctly discriminate the observations for best classification results rather than to fit the distributions to the data [100]. This training criterion is normally based on a smooth function of the difference between the log-likelihood of the correct word sequence and all other competing sequences, and a sigmoid function is often used [66]. MCE training can be described as minimising:

$$\mathcal{F}_{\text{MCE}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \left(1 + \left[\frac{P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda})}{\sum_{W \neq W_n} P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})} \right]^{\sigma} \right)^{-1} \tag{2.18}$$

where $P(W_n|\boldsymbol{O}_n, \boldsymbol{\lambda})$ is defined in equation (2.17), and σ is an additional smoothing term introduced by the sigmoid smoothing function. Compared with MMI estimation, in the MCE training criterion, the denominator term does not include the correct word sequence, and a sigmoid smoothing function is used [66]. When the smoothing term $\sigma = 1$, then yields:

$$\mathcal{F}_{\text{MCE}}(\boldsymbol{\lambda}) = N - \sum_{n=1}^{N} P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda}) \tag{2.19}$$

This is one specification of the minimum Bayes risk criterion, which will be discussed in the following section.

2.3.4 Minimum Bayes Risk (MBR)

In *minimum Bayes risk* (MBR) training, the expected loss in recognition is aimed to be minimised [74, 103]. Normally, the expected loss estimated on the training data is used as

an approximation [66]. MBR training then can be described as minimising the following objective function:

$$\mathcal{F}_{\text{MBR}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \sum_{W} P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})\mathcal{L}(W, W_n) \tag{2.20}$$

where $\mathcal{L}(W, W_n)$ is the loss function, that measures how different the word sequence W and the reference W_n are. There are a number of definitions for the loss function, and these definitions lead to different meaningful training criteria which will be discussed in the following paragraphs:

- **1/0 loss**: For continuous speech recognition, 0/1 loss is equivalent to a sentence-level loss function. The loss function can be defined as follows:

$$\mathcal{L}(W, W_n) = \begin{cases} 1; & W \neq W_n \\ 0; & W = W_n \end{cases} \tag{2.21}$$

  When $\sigma = 1$, the MCE training criterion (2.19) is the same as the MBR training criterion (2.20) with 1/0 loss defined in (2.21).

- **Word-level loss**: This loss function is directly related to the expected word error rate (WER). It is normally computed by minimising the Levenshtein edit distance [112] between the word sequences W and W_n. By using this word-level loss, MBR training is known as *minimum word error* (MWE) training [120].

- **Phone-level loss**: For large vocabulary speech recognition not all word sequences will be observed. To ensure generalisation, the loss function is often computed between phone sequences, rather word sequences [66]. This is known as *minimum phone error* (MPE) training [140, 141].

- **Frame-level loss**: Compared to the number of frames, the use of phone-level loss function reduces the number of possible errors to be corrected, and this might cause generalisation issues [66]. To address this problem, *minimum phone frame error* (MPFE) training was proposed, where the loss is defined as a measure of the number of frames having incorrect phone labels [211]. This is the same as the Hamming distance [173].

The MMI, MCE and MPE criteria have been compared on the Wall Street Journal (WSJ) task in [118]. In recognition, all these discriminative training significantly outperformed ML

training in terms of WER, and both MCE and MPE were found to outperform MMI on this task. In work [211], MPFE was reported to give small but consistent gains over MPE.

2.3.5 Large Margin Training

Alternative to MMI, MCE and MBR training, large margin training has been successfully used in speech recognition [94, 107, 113, 161]. For a large margin classifier, the generalisation error is bounded by the sum of the training error and a term that depends on the *Vapnik-Chervonenkis* (VC) dimension [34, 183]. The margin is the smallest distance between the reference label (correct class) and any alternative label (incorrect class), and for sequential data it can be expressed in the form of the log-posterior ratio [136, 147, 164]. The simplest form of the large margin training criterion can be described as maximising:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \left[\min_{W \neq W_n} \left\{ \log \left(\frac{P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda})}{P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})} \right) \right\} \right] \tag{2.22}$$

where the class posterior distributions $P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda})$ and $P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})$ are defined in equation (2.17), and the normalisation terms of the posterior distributions can be cancelled out in the large margin training criterion (2.22).

For a sequence classification task it is important to take into account loss. Alternative to criterion (2.22), in work [161] the margin is defined as being not smaller than a loss function. By using this type of margin definition, large margin training can be described as minimising the following objective function [147]:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \left[\max_{W \neq W_n} \left\{ \mathcal{L}(W, W_n) - \log \left(\frac{P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda})}{P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})} \right) \right\} \right]_+ \tag{2.23}$$

where $\mathcal{L}(W, W_n)$ is the loss function, and the forms of this loss function were discussed in section 2.3.4. In order to omit the data that have already been classified correctly and beyond the margin, the hinge loss function $\left[\,\cdot\,\right]_+$ is introduced in the large margin training criterion (2.23), and it is defined as follows:

$$\left[f(x)\right]_+ = \begin{cases} 0 & \text{when } f(x) < 0 \\ f(x) & \text{when } f(x) \geq 0 \end{cases} \tag{2.24}$$

Because of the $\max\{\cdot\}$ function, the objective function described in (2.23) is not differentiable. In order to simplify optimisation, the following soft-max inequality can be used as

an approximation:

$$\max_i\{x_i\} \leq \log\left(\sum_i \exp(x_i)\right) \tag{2.25}$$

By using this inequality (2.25), the large margin training criterion (2.23) can be relaxed to its upper bound:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\lambda}) \leq \sum_{n=1}^{N}\left[-\log P(W_n|\boldsymbol{O}_n;\boldsymbol{\lambda}) + \log\left(\sum_W P(W|\boldsymbol{O}_n;\boldsymbol{\lambda})\mathcal{L}_{\exp}(W,W_n)\right)\right]_+ \tag{2.26}$$

where the loss function $\mathcal{L}_{\exp}(W,W_n)$ is defined as [147]:

$$\mathcal{L}_{\exp}(W,W_n) = \begin{cases} \exp\left(\mathcal{L}_{\exp}(W,W_n)\right) & \text{when } W \neq W_n \\ 0 & \text{when } W = W_n \end{cases} \tag{2.27}$$

This upper bound (2.26) is related to the MMI and MBR objective functions described in (2.16) and (2.20) respectively. The first term within the hinge loss function is the negated log-posterior, which is the same as the MMI objective function; The second term is the logarithm of a MBR variant, where the loss function is defined as in equation (2.27) [147]. Furthermore, if the conditional probability $P(W|\boldsymbol{O};\boldsymbol{\lambda})$ is written in the form consisting of acoustic model likelihood and language model probability as described in (2.17), this upper bound (2.26) is related to the boosted MMI (BMMI) criterion [143, 153]. The BMMI objective function to be maximised can be described as:

$$\mathcal{F}_{\text{BMMI}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N}\log\left(\frac{p(\boldsymbol{O}_n|W_n;\boldsymbol{\lambda})P(W_n)}{\sum_W p(\boldsymbol{O}_n|W;\boldsymbol{\lambda})P(W)\exp\left(-b\mathcal{A}(W,W_n)\right)}\right) \tag{2.28}$$

where b is a boosting factor. $\mathcal{A}(W,W_n)$ is the accuracy between a word sequence W and the given reference W_n, and this accuracy can be expressed in terms of the number of correct phones in W as in MPE [153]. There are two main differences between the upper bound (2.26) and the BMMI objective function (2.28): in the former the hinge loss function is used, and in the latter a scaled accuracy function is used instead of the loss function (2.27) [147].

2.4 Adaptation and Adaptive Training

In speech recognition, since there will always be new speakers and new environments, it is common that the training data cannot adequately represent the test data, and this mismatch

might significantly degrade the recognition performance [66]. To address this issue, adaptation was proposed to compensate the mismatch of acoustic conditions between the training and test data. Adaptation allows a small amount of data from target speaker (or noise condition) to be used to transform an acoustic model set to make it more closely match that speaker (or noise condition). By using adaptation, significant improvement can be achieved on the test data with various acoustic conditions [66, 203]. In general, adaptation can be performed by feature based approaches, model based approaches or combinations of them. Feature based approaches can be used to normalise acoustic features such that the mismatch between the training and test data can be reduced, whereas model based approaches are generally more powerful as they have more modelling power to represent acoustic variability and handle uncertainty [56, 192]. In the following subsections, the commonly used model based schemes such as *maximum a posteriori* (MAP) and *maximum likelihood linear regression* (MLLR) will be introduced. Many of these model based schemes can be employed in the framework of adaptive training, where speech variability and non-speech variability (such as speaker or acoustic conditions) are modelled separately [203].

2.4.1 Maximum a Posteriori (MAP)

The adaptation data can be viewed as additional training data, hence the most straightforward way to perform adaptation is to re-estimate the model parameters based on ML training. However, this approach might be problematic, since the amount of adaptation data usually is small. This leads to over-fitting of the trained model. To address this problem, *maximum a posteriori* (MAP) estimation was proposed [68], where in addition to the adaptation data, a prior distribution over the model parameters is used in parameter estimation. Let the model parameters be $\boldsymbol{\lambda}$, given the adaptation data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, MAP estimation can be described as maximising the following objective function:

$$\begin{aligned} \mathcal{F}_{\text{MAP}}(\boldsymbol{\lambda}) &= \log p(\boldsymbol{\lambda}) + \mathcal{F}_{\text{ML}}(\boldsymbol{\lambda}) \\ &= \log p(\boldsymbol{\lambda}) + \sum_{n=1}^{N} \log p(\boldsymbol{O}_n | W_n; \boldsymbol{\lambda}) \end{aligned} \tag{2.29}$$

where $p(\boldsymbol{\lambda})$ is the prior distribution over the model parameters, and $\mathcal{F}_{\text{ML}}(\boldsymbol{\lambda})$ is the objective function of maximum likelihood estimation described in (2.14). By using MAP adaptation, the original prior parameter values can be effectively interpolated with those that would be

obtained from the adaptation data alone. This makes MAP especially useful for porting a well-trained model set to a new domain where only a limited amount of data is available [66]. One major drawback of MAP adaptation is that each Gaussian component is updated individually. If the adaptation data is sparse, then many of the model parameters will not be updated [66]. To address this problem, various extensions of MAP estimation have been proposed, e.g. the regression based model prediction [3] and structured MAP [163]. Alternative to extensions within the MAP framework, linear transform based approaches can be adopted to perform rapid adaptation of all Gaussian parameters, which will be discussed in the following subsection.

2.4.2 Linear Transform Based Adaptation

In speech recognition, the linear transform based adaptation approaches are widely used. Especially when the amount of adaptation data is limited, this type of adaptation is currently the most effective form [66]. In the transform based adaptation approaches, a set of linear transforms are used to map the existing model to a new adapted one such that the likelihood of the model parameters is maximised given the adaptation data [66]. When a single global transform $\boldsymbol{T}$ is considered, given the adaptation data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$ and the unadapted model parameters $\bar{\boldsymbol{\lambda}}$, the transform then can be estimated by maximising the following likelihood function:

$$\mathcal{F}(\boldsymbol{T}) = \sum_{n=1}^{N} p(W_n|\boldsymbol{O}_n; \bar{\boldsymbol{\lambda}}, \boldsymbol{T}) \tag{2.30}$$

In this thesis, three types of commonly used linear transform based adaptation will be introduced, namely *maximum likelihood linear regression* (MLLR) [110], *variance MLLR* and *constrained MLLR* (CMLLR) [55].

In the following subsections, bar notation will be used to denote the unmodified (or canonical) acoustic models and unmodified (or "clean") observations. For example, $\bar{\boldsymbol{\lambda}}$ denotes the canonical set of HMM parameters, whilst $\boldsymbol{\lambda}$ denotes the adapted set of HMM parameters. Similarly, $\bar{\boldsymbol{o}}$ denotes the "clean" observation, whilst $\boldsymbol{o}$ denotes the noise-corrupted observation.

2.4.2.1 Maximum Likelihood Linear Regression (MLLR)

Maximum likelihood linear regression (MLLR) [110] was originally proposed to transform the mean vectors of the Gaussian components. Transforms later were extended to the covariance matrices [55]. In *mean MLLR*, only the means of the Gaussian components are transformed. For the mth component, the transform can be described as [110]:

$$\boldsymbol{\mu}_m = \boldsymbol{A}\bar{\boldsymbol{\mu}}_m + \boldsymbol{b} \tag{2.31}$$

where $\{\boldsymbol{A}, \boldsymbol{b}\}$ are the transform parameters associated with the mean vectors. In addition to mean vectors, transforms can also be applied to the covariance matrices. Then the transform can be described as follows [55]:

$$\boldsymbol{\Sigma}_m = \boldsymbol{H}\bar{\boldsymbol{\Sigma}}_m\boldsymbol{H}^{\mathsf{T}} \tag{2.32}$$

where $\boldsymbol{H}$ are the transform parameters associated with the covariance matrices. This type of transform is usually called *variance MLLR* [200]. When both the mean vectors and covariance matrices are adapted, the likelihood for the mth Gaussian component can be computed by transforming the observations and means whilst keeping the covariance matrices unchanged [66]:

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{|\boldsymbol{H}|}\mathcal{N}\big(\boldsymbol{H}^{-1}\boldsymbol{o}; \boldsymbol{H}^{-1}(\boldsymbol{A}\bar{\boldsymbol{\mu}}_m + \boldsymbol{b}), \bar{\boldsymbol{\Sigma}}_m\big) \tag{2.33}$$

When using this form (2.33), the likelihood can be efficiently computed by caching the transformed observations and means, especially in situations when the covariance matrices are diagonal [66, 200].

2.4.2.2 Constrained MLLR

In *constrained MLLR* (CMLLR), both the mean vectors and covariance matrices of the Gaussian components are transformed, and the transform matrices $\boldsymbol{A}$ and $\boldsymbol{H}$ are constrained to be the same. Then the CMLLR transform can be described as follows [55]:

$$\boldsymbol{\mu}_m = \boldsymbol{A}\bar{\boldsymbol{\mu}}_m + \boldsymbol{b} \tag{2.34}$$

$$\boldsymbol{\Sigma}_m = \boldsymbol{A}\bar{\boldsymbol{\Sigma}}_m\boldsymbol{A}^{\mathsf{T}} \tag{2.35}$$

where $\boldsymbol{A}$ is the constrained transform parameters, and $\boldsymbol{b}$ is the bias for the mean transform. With this transform constraint, the likelihood described in (2.33) can be further written as:

$$\begin{aligned}\mathcal{N}(\boldsymbol{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) &= \frac{1}{|\boldsymbol{A}|}\mathcal{N}\left(\boldsymbol{A}^{-1}\boldsymbol{o}-\boldsymbol{A}^{-1}\boldsymbol{b};\bar{\boldsymbol{\mu}}_m,\bar{\boldsymbol{\Sigma}}_m\right)\\ &= \frac{1}{|\boldsymbol{A}|}\mathcal{N}\left(\bar{\boldsymbol{o}};\bar{\boldsymbol{\mu}}_m,\bar{\boldsymbol{\Sigma}}_m\right)\end{aligned} \tag{2.36}$$

where $\bar{\boldsymbol{o}}$ is the transformed observation:

$$\bar{\boldsymbol{o}} = \boldsymbol{A}^{-1}\boldsymbol{o}-\boldsymbol{A}^{-1}\boldsymbol{b} \tag{2.37}$$

Compared with mean and variance MLLR, CMLLR can be operated in the form of transforming observations as described in (2.37). This makes this type of transform efficient if the speaker (or environment) changes rapidly [66]. CMLLR is the form of linear transform most frequently used in adaptive training, which will be discussed at the end of this chapter. For the detail of transform estimation on the mean, variance and constrained MLLR, consulting the references [55, 110].

2.4.3 Vector Taylor Series (VTS)

The linear transform based schemes (discussed in the previous subsection) is usually used to adapt a speech recognition system to changes in speaker. Although these methods can reduce the effect of noise, more effective approaches have been proposed to compensate for noise effect, and these approaches are normally referred to as *noise compensation*. Adaptation and compensation have become similar in recent years and share many of the same attributes [192], hence in this thesis these two terms are used interchangeably. In this subsection, a noise compensation approach called *vector Taylor series* (VTS) compensation [2, 128] will be introduced.

Consider a simplified noisy acoustic environment model, where the clean speech signal $\bar{o}$ (in the time domain) is corrupted by additive noise n and channel distortion h as illustrated in Figure 2.5. In the time domain, the relationship between the noise corrupted and clean speech signals (or the *mismatch function*) can be described as follows:

$$o = \bar{o}\otimes h + n \tag{2.38}$$

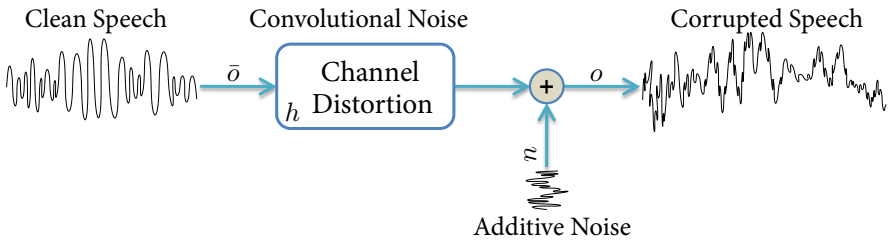


Figure 2.5: *A simplified model of noisy acoustic environment.*

where $\otimes$ denotes convolution in the time domain. In the HMM based recognition system, the *Mel-frequency cepstral coefficients* (MFCC) [89, 200] are widely used. In the MFCC domain, the mismatch function (2.38) can be described as follows[1] [1, 2, 63]:

$$\begin{aligned}\boldsymbol{o}^{\mathrm{s}} =& \bar{\boldsymbol{o}}^{\mathrm{s}} + \boldsymbol{h}^{\mathrm{s}} + \boldsymbol{C}\log\left(1+\exp\left(\boldsymbol{C}^{-1}(\boldsymbol{n}^{\mathrm{s}} - \bar{\boldsymbol{o}}^{\mathrm{s}} - \boldsymbol{h}^{\mathrm{s}})\right)\right) \\ =& \bar{\boldsymbol{o}}^{\mathrm{s}} + f(\bar{\boldsymbol{o}}^{\mathrm{s}}, \boldsymbol{n}^{\mathrm{s}}, \boldsymbol{h}^{\mathrm{s}})\end{aligned} \tag{2.39}$$

where $\boldsymbol{o}^{\mathrm{s}}$ and $\bar{\boldsymbol{o}}^{\mathrm{s}}$ are the noise corrupted and clean (static) observations. The superscript $^{\mathrm{s}}$ denotes the static coefficients. In speech recognition, an observation $\boldsymbol{o}$ is often comprised of static coefficients appended with delta (Δ) and delta-delta (Δ^2) dynamic coefficients [52, 66], namely $\boldsymbol{o} = \left[\boldsymbol{o}^{\mathrm{s}\mathsf{T}}, \Delta\boldsymbol{o}^{\mathrm{s}\mathsf{T}}, \Delta^2\boldsymbol{o}^{\mathrm{s}\mathsf{T}}\right]^{\mathsf{T}}$. In the mismatch function (2.39), $\boldsymbol{n}^{\mathrm{s}}$ is the additive noise, $\boldsymbol{h}^{\mathrm{s}}$ is the convolutional noise or channel distortion, and $\boldsymbol{C}$ is the *discrete cosine transform* (DCT) matrix.

Model compensation is aimed to obtain the parameters of the noise corrupted speech model from the clean speech and noise models [63]. Many of the model compensation approaches assume that if the clean speech and noise models are Gaussian, namely $\mathcal{N}(\bar{\boldsymbol{\mu}}^{\mathrm{s}}, \bar{\boldsymbol{\Sigma}}^{\mathrm{s}})$, $\mathcal{N}(\boldsymbol{\mu}^{\mathrm{s,n}}, \boldsymbol{\Sigma}^{\mathrm{s,n}})$ and $\mathcal{N}(\boldsymbol{\mu}^{\mathrm{s,h}}, \boldsymbol{\Sigma}^{\mathrm{s,h}})$[2], then the noise corrupted speech model is also Gaussian. Thus, the parameters of the corrupted speech distribution $\mathcal{N}(\boldsymbol{\mu}^{\mathrm{s}}_m, \boldsymbol{\Sigma}^{\mathrm{s}}_m)$ for a particular component can be written as:

$$\boldsymbol{\mu}^{\mathrm{s}}_m = \mathcal{E}\{\boldsymbol{o}^{\mathrm{s}}\} \tag{2.40}$$

$$\boldsymbol{\Sigma}^{\mathrm{s}}_m = \mathcal{E}\{\boldsymbol{o}^{\mathrm{s}}\boldsymbol{o}^{\mathrm{s}\mathsf{T}}\} - \boldsymbol{\mu}^{\mathrm{s}}_m\boldsymbol{\mu}^{\mathrm{s}\mathsf{T}}_m \tag{2.41}$$

where the expectation is taken with respect to the component distribution of the clean speech model and the distribution of the noise model. The relationship between the noise

[1] In this section, when applying $\log(\cdot)$ or $\exp(\cdot)$ to a vector, an element-wise logarithm or exponential function is performed to all elements of the vector.

[2] The convolution noise is usually assumed to be constant, namely $\boldsymbol{\Sigma}^{\mathrm{s,h}} = 0$ [2, 63].

corrupted and clean (static) speech ($\boldsymbol{o}^{\mathrm{s}}$ and $\bar{\boldsymbol{o}}^{\mathrm{s}}$) is described in equation (2.39). According to this equation (2.39), the noisy corrupted (static) speech $\boldsymbol{o}^{\mathrm{s}}$ is a highly non-linear function of the underlying clean (static) speech $\bar{\boldsymbol{o}}^{\mathrm{s}}$. There are no simple closed form solutions to these expectations in equations (2.40) and (2.41), hence various approximate approaches have been proposed. *Parallel model combination* (PMC) [61, 62] is one such approach, where the Gaussian means and variances in the cepstral domain are mapped into a linear domain in which the noise is additive, and the means and variances of the new distribution for the noise corrupted speech are computed, then they are mapped back to the cepstral domain [66]. An alternative approximate approach is VTS compensation [2, 128], where the vector Taylor series expansion is used to dealing with non-linearity.

In VTS compensation [2], the non-linearity between the noise corrupted speech and clean speech can be represented by using the first order vector Taylor series expansion of equation (2.39). For a particular component, when the expansion points for the vector Taylor series are the means of the clean speech, additive noise and convolutional noise (namely $\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m$, $\boldsymbol{\mu}^{\mathrm{s,n}}$ and $\boldsymbol{\mu}^{\mathrm{s,h}}$), then the mean of the noise corrupted speech distribution can be described as follows:

$$\boldsymbol{\mu}^{\mathrm{s}}_m = \mathcal{E}\Big\{\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m + f(\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m, \boldsymbol{\mu}^{\mathrm{s,n}}, \boldsymbol{\mu}^{\mathrm{s,h}}) + (\boldsymbol{o}^{\mathrm{s}} - \bar{\boldsymbol{\mu}}^{\mathrm{s}}_m)\frac{\partial f}{\partial \boldsymbol{o}^{\mathrm{s}}} + (\boldsymbol{n}^{\mathrm{s}} - \boldsymbol{\mu}^{\mathrm{s,n}})\frac{\partial f}{\partial \boldsymbol{n}^{\mathrm{s}}} + (\boldsymbol{h}^{\mathrm{s}} - \boldsymbol{\mu}^{\mathrm{s,h}})\frac{\partial f}{\partial \boldsymbol{h}^{\mathrm{s}}}\Big\} \tag{2.42}$$

where function $f(\cdot)$ is defined in equation (2.39), and the expectation is taken over the clean speech and noise distributions, namely $\mathcal{N}(\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m, \bar{\boldsymbol{\Sigma}}^{\mathrm{s}}_m)$ for the clean speech, $\mathcal{N}(\boldsymbol{\mu}^{\mathrm{s,n}}, \boldsymbol{\Sigma}^{\mathrm{s,n}})$ for the additive noise, and $\mathcal{N}(\boldsymbol{\mu}^{\mathrm{s,h}}, \boldsymbol{\Sigma}^{\mathrm{s,h}})$ for the convolutional noise. The partial derivatives in (2.42) can be expressed in terms of partial derivatives of $\boldsymbol{o}^{\mathrm{s}}$ with respect to $\bar{\boldsymbol{o}}^{\mathrm{s}}$, $\boldsymbol{n}^{\mathrm{s}}$ and $\boldsymbol{h}^{\mathrm{s}}$ evaluated at $\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m$, $\boldsymbol{\mu}^{\mathrm{s,n}}$ and $\boldsymbol{\mu}^{\mathrm{s,h}}$, and they can be described as follows [2, 66]:

$$\partial f/\partial \boldsymbol{o}^{\mathrm{s}} = \partial f/\partial \boldsymbol{h}^{\mathrm{s}} = \boldsymbol{A} \tag{2.43}$$

$$\partial f/\partial \boldsymbol{n}^{\mathrm{s}} = \boldsymbol{I} - \boldsymbol{A} \tag{2.44}$$

where $\boldsymbol{A} = \boldsymbol{C}\boldsymbol{F}\boldsymbol{C}^{-1}$ and $\boldsymbol{F}$ is a diagonal matrix whose elements are given by $1/\big(1 + \exp(\boldsymbol{C}^{-1}(\boldsymbol{n}^{\mathrm{s}} - \bar{\boldsymbol{o}}^{\mathrm{s}} - \boldsymbol{h}^{\mathrm{s}}))\big)$. With the first order vector Taylor series approximation (2.42), the mean and variance of the noisy corrupted speech can be described as follows [2, 66]:

$$\boldsymbol{\mu}^{\mathrm{s}}_m = \bar{\boldsymbol{\mu}}^{\mathrm{s}}_m + f(\bar{\boldsymbol{\mu}}^{\mathrm{s}}_m, \boldsymbol{\mu}^{\mathrm{s,n}}, \boldsymbol{\mu}^{\mathrm{s,h}}) \tag{2.45}$$

$$\boldsymbol{\Sigma}^{\mathrm{s}}_m = \boldsymbol{A}\bar{\boldsymbol{\Sigma}}^{\mathrm{s}}_m\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{A}\boldsymbol{\Sigma}^{\mathrm{s,h}}\boldsymbol{A}^{\mathsf{T}} + (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{\Sigma}^{\mathrm{s,n}}(\boldsymbol{I} - \boldsymbol{A})^{\mathsf{T}} \tag{2.46}$$

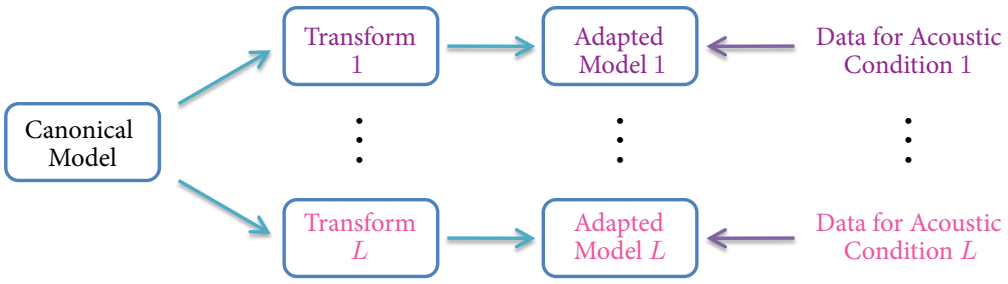


Figure 2.6: *The framework of linear transform based adaptive training.*

In practice, the convolution noise is usually assumed to be constant, namely $\boldsymbol{\Sigma}^{\mathrm{s,h}} = 0$. Normally $\boldsymbol{\mu}^{\mathrm{s,n}}$, $\boldsymbol{\Sigma}^{\mathrm{s,n}}$ and $\boldsymbol{\mu}^{\mathrm{s,h}}$ are seldom known in advance, so they must be estimated from the test data. When they are not available, noise estimation can be based on the maximum likelihood (ML) noise estimation scheme [115]. For compensation of the delta and delta-delta coefficients, the *continuous time approximation* [76] is commonly used, where discrete time estimation of the gradient is approximated by the derivative with respect to time. More details can be found in [2].

2.4.4 *Adaptive Training*

In adaptation, a well trained acoustic model is pre-required, and the traditional approach to obtain the model is to train it on the data from a single source [66, 203]. The model then can be adapted to the test domain during recognition using the adaptation techniques introduced in the previous subsections. However, normally the the training data includes a large number of acoustic conditions such as speakers or noise conditions, hence acoustic mismatches exit within the training data. One approach to handling this problem is to use adaptation during training, and this type of method is referred to as *adaptive training* [5, 66], in which speech variability and acoustic conditions are modelled separately. Thus, in adaptive training (take the linear transform based approach for example), two sets of models are obtained, namely the canonical model $\bar{\boldsymbol{\lambda}}$ (which represents speech variability) and the transforms $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_L\}$ (which represent different acoustic conditions) , where L is the number of different acoustic conditions. The framework of linear transform based adaptive training is illustrated in Figure 2.6. This type of adaptive training is also known as *speaker adaptive training* (SAT) [5], given that it was first proposed to handle speaker variability, where each speaker corresponds to an acoustic condition. The training procedure

of adaptive training is performed in an iterative way, and this process can be summarised as follows [66]:

(1) Initialise the canonical model and the transform for each acoustic condition.

(2) Estimate the transform for each acoustic condition using the training data associated with that condition.

(3) Estimate the canonical model given all of the transforms.

(4) Goto step (2) until convergence or the maximum number of iterations is reached.

The canonical model obtained from adaptive training cannot be directly employed in speech recognition, and the transforms for the test data need to be estimated given some supervision data. Then, the adapted model (the canonical model with transforms) can be used in the final recognition. Since CMLLR is simple to implement, it is widely used in adaptive training [58].

2.5 Summary

In this chapter, the most commonly used generative models in speech recognition, such Gaussian mixture models (GMMs) and hidden Markov models (HMMs), were introduced. In speech recognition, the state-of-the-art systems often employ deep neutral networks (DNNs), therefore the frameworks of the DNN-HMM system and the joint decoding system were briefly introduced. Various training criteria for generative models, such as maximum likelihood (ML), maximum mutual information (MMI), minimum Bayes risk (MBR) and large margin training, were also briefly discussed. The mismatch between the training and test data is a common issue in speech recognition, hence the techniques such as adaptation and compensation to reduced this mismatch were discussed in this chapter. Moreover, adaptive training, which deals with the problem of the mismatch within the training data, was briefly discussed.

Chapter 3

Discriminative Models

In the previous chapter, the most commonly used generative models in speech recognition were introduced. To adopt a generative model in the classification task, Bayes' rule needs to be applied to yield the posterior distribution of the class label given the observations. In speech recognition, HMMs are the most widely used acoustic models. Though discriminative training of the HMM normally achieves performance gains compared with generative training, however the underlying model is still generative. Recently, applying discriminative models directly to speech recognition is of growing interest [48, 54, 65, 79, 106, 215], where the conditional distribution of the class label given the observations is modelled directly. Compared with generative models, discriminative models may lead to improved performance, particularly when the class-conditional density assumptions of the generative models give a poor approximation to the true distributions [16]. Moreover, discriminative models do not make any assumption on the distribution of the input data, and not need to model the density as an incremental step, but focus on the boundary between classes. This means discriminative models do not waste any resources trying to model the joint distribution [132, 183]. As illustrated in Figure 3.1, the complicated structure in the probability density might have little impact on the posterior probabilities. Therefore, it is not always desirable to compute the joint distribution. Finally, discriminative models have the potential to improve performance as a wider range of features from the observation and word sequences can be used in training compared with generative models [54]. These are the main reasons why discriminative models have been widely and successfully used [105, 132, 183].

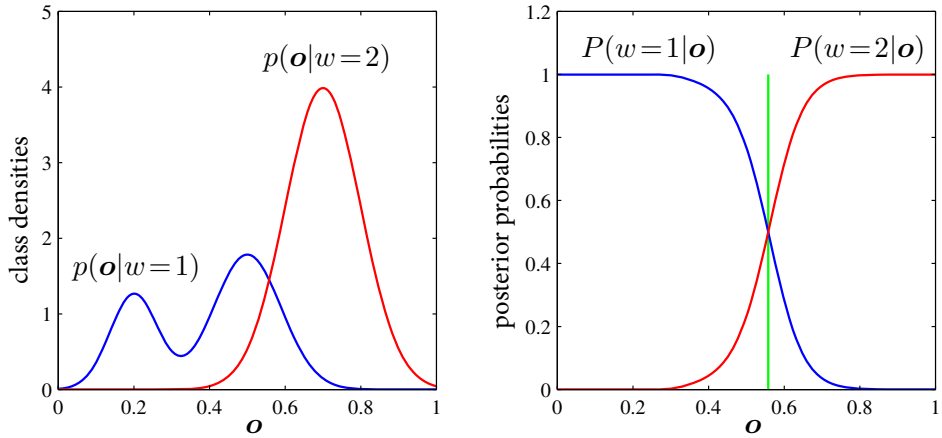


Figure 3.1: *The generative model versus the discriminative model. This figure is taken from [16]. Example of the class-conditional densities for 2 classes having a single input variable* $\boldsymbol{o}$ *(left plot) together with the corresponding posterior probabilities (right plot). Note that the left-hand mode of the class conditional density* $p(\boldsymbol{o}|w=1)$ *shown in blue on the left plot, has no effect on the posterior probabilities. The vertical green line in the right plot shows the decision boundary in* $\boldsymbol{o}$ *that gives the minimum misclassification rate.*

In this chapter, the most commonly used discriminative models will be introduced. Various training criteria and forms of features for discriminative models also will be discussed.

In general, discriminative models can be divided into two groups, namely the unstructured[1] and structured discriminative models. For structured discriminative models, the class labels are sequences and different class labels share the same common units. For example, in speech recognition sentences are structured labels, and different sentences share the same common units of words (or phones). For unstructured discriminative models, the class labels are single (atomic) units, and different class labels distinguish from each other. In the following sections, these two types of discriminative models will be discussed in detail.

3.1 Unstructured Discriminative Models

In this section, some of the commonly used unstructured discriminative models will be introduced. In order to distinguish from the class label W which is a sequence, in this

[1] In some literatures, the unstructured model is called the flat model.

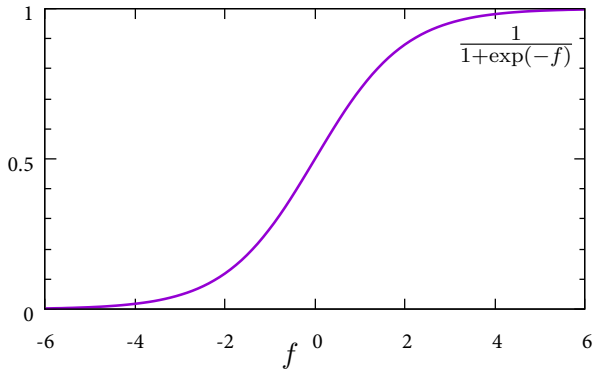


Figure 3.2: *The logistic sigmoid function.*

thesis w is used as the notation of the class label for unstructured discriminative models. The unstructured class labels are single units. For example, if the number of classes is L, the class label w takes value from $\{1, \ldots, L\}$. Unstructured discriminative models can be directly applied to isolated word recognition [14, 63] or phone classification [152]. However, in continuous speech recognition (CSR) the number of the possible classes for an utterance is exponentially large. One solution to this problem is to use *acoustic code breaking* [185], which will be discussed at the end of this section.

3.1.1 Logistic Regression

For a discriminative model, the conditional distribution $P(w|\boldsymbol{O})$ is modelled directly, where $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ is a sequence of observations. In a two-class (or binary) classification problem with class labels $\{-1, 1\}$, the conditional probability $P(w|\boldsymbol{O})$ for class 1 can be written as a logistic sigmoid acting on a linear function of the feature vector [16]:

$$P\big(w=1|\boldsymbol{O}\big) = \frac{1}{1+\exp(-f)} \tag{3.1}$$

The graphical representation of the logistic sigmoid function is illustrated in Figure 3.2. In equation (3.1), f is a linear function of the feature vector $\varphi(\boldsymbol{O})$:

$$f = \boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}) \tag{3.2}$$

Given the form of the conditional distribution (3.1) for class 1, the conditional distribution for class -1 can be written as $P\big(w=-1|\boldsymbol{O}\big) = 1 - P\big(w=1|\boldsymbol{O}\big)$. This type of model is

known as *logistic regression*[1] [32, 189]. In (3.2), $\boldsymbol{\eta}$ are the model parameters, and $\varphi(\cdot)$ is the feature function that maps the observations $\boldsymbol{O}$ with various length to a fixed dimensional space. One example of the feature vector is [209]:

$$\varphi(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \sum_{t=1}^{T} \boldsymbol{o}_t \end{bmatrix} \tag{3.3}$$

where 1 is introduced to allow a bias parameter.

Logistic regression only can be used in binary classification. In order to extend to multi-class classification, the conditional probability $P(w|\boldsymbol{O})$ can be given by a softmax transformation of a linear function of the feature vector [16]:

$$P(w|\boldsymbol{O}) = \frac{\exp(f_w)}{\sum_w \exp(f_w)}, \quad w \in \{1, \ldots, L\} \tag{3.4}$$

where $\{1, \ldots, L\}$ are L different classes, and f_w is a linear function of the feature vector:

$$f_w = \boldsymbol{\eta}_w^{\mathsf{T}} \varphi(\boldsymbol{O}) \tag{3.5}$$

where $\boldsymbol{\eta}_w$ are the model parameters associated with class w. This type of model (3.4) is referred to as *multinomial logistic regression*, which is also known as the *maximum entropy model* [10, 119], (unstructured) *log-linear model*, or single-layer *artificial neural network* (ANN) [15]. Given that a valid probability distribution satisfies $\sum_w P(w|\boldsymbol{O}) = 1$, the parameters for one class, such as $\boldsymbol{\eta}_w$ do not need to be estimated and can be set to $\mathbf{0}$. However, these redundant parameters are normally kept in optimisation, for the numerical stability reasons and equal treatment of all classes [14, 171].

The parameters $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_L\}$ of multinomial logistic regression can be combined to be a single vector [209]:

$$\boldsymbol{\eta} = \left[\boldsymbol{\eta}_1^{\mathsf{T}}, \ldots, \boldsymbol{\eta}_L^{\mathsf{T}}\right]^{\mathsf{T}} \tag{3.6}$$

Then the linear function of the feature vector f_w described in (3.5) can be expressed as:

$$f_w = \boldsymbol{\eta}_w^{\mathsf{T}} \varphi(\boldsymbol{O}) = \boldsymbol{\eta}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{O}, w) \tag{3.7}$$

where $\boldsymbol{\phi}(\boldsymbol{O}, w)$ is a sparse feature vector that characterises the dependencies between observations $\boldsymbol{O}$ and class w:

$$\boldsymbol{\phi}(\boldsymbol{O}, w) = \begin{bmatrix} \delta(w, 1)\varphi(\boldsymbol{O}) \\ \vdots \\ \delta(w, L)\varphi(\boldsymbol{O}) \end{bmatrix} \tag{3.8}$$

[1] It is worth noting that logistic regression is a model for classification rather than regression.

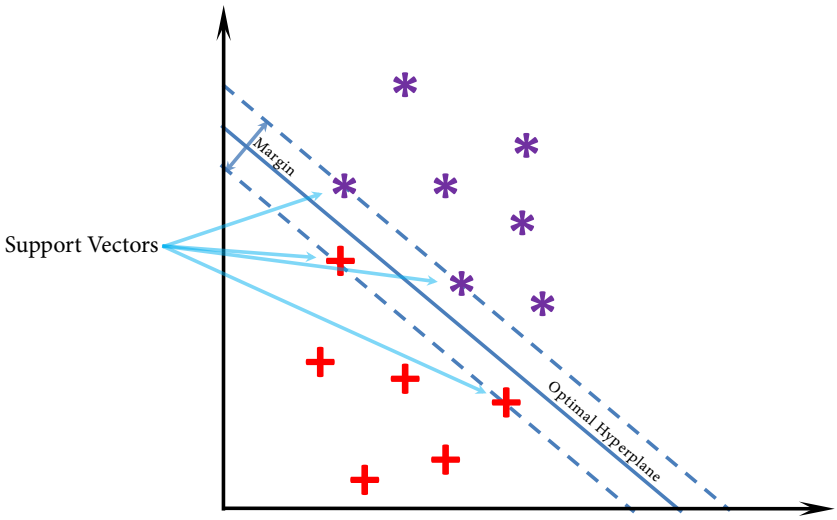


Figure 3.3: *Binary classification using the SVM for separable data.*

where $\delta(\cdot)$ is the Kronecker delta, and $\varphi(\cdot)$ is a feature function, e.g. the example given in equation (3.3). More forms of the feature function $\varphi(\cdot)$ will be discussed in section 3.5.

3.1.2 Support Vector Machines

Support Vector Machines (SVMs) [31, 183] are widely used supervised learning model in (binary) classification. For SVMs, the generalisation error is bounded by the sum of the training error and a term that depends on the Vapnik-Chervonenkis (VC) dimension [34, 108, 183]. Good generalisation of this type of classifier has enabled it to be successfully used in various research fields including speech recognition [60, 63, 67]. Intuitively, the SVM constructs a hyperplane that has the largest distance to the nearest training data point of any class. This distance between classes is referred to as margin. Figure 3.3 illustrates a binary classification task using the SVM for separable data in the two-dimensional space.

Consider a training set consisting of feature vector and class label pairs $\mathcal{D} = \{(\varphi(\boldsymbol{O}_1), w_1), \ldots, (\varphi(\boldsymbol{O}_N), w_N)\}$, where $w_n \in \{-1, +1\}$ and $\varphi(\boldsymbol{O}_n)$ is the feature vector for observations $\boldsymbol{O}_n$. For a separable binary classification task, given the hyperplane of the SVM, which are parameterised by $\boldsymbol{\eta}$ and bias b, correct classification of the training data will satisfy:

$$w_n\left(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}_n) + b\right) \geq 1, \quad \forall n \in \{1, \ldots, N\} \tag{3.9}$$

where the training data are constrained to be not inside the region enclosed by the margin edges. The data located at the edges, namely $w_n\left(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}_n)+b\right)=1$, are called *support vectors*. As illustrated in Figure 3.3, the dashed lines are margin edges, and the solid line is the hyperplane (decision boundary). The margin is the distance between the dashed edges, and it can be expressed as [31]:

$$\text{margin}=\frac{2}{||\boldsymbol{\eta}||} \tag{3.10}$$

The optimal hyperplane is the unique one that maximises the margin (3.10) under the constraints in (3.9), hence constructing an optimal hyperplane (decision boundary) is a quadratic problem, and this optimal hyperplane is estimated such that the margin is maximised and all the training data are correctly classified[1] [31]:

$$\begin{aligned}&\min_{\boldsymbol{\eta},b}\frac{1}{2}||\boldsymbol{\eta}||^2 \\ &\text{s.t.}\quad w_n\left(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}_n)+b\right)\geq 1,\quad \forall n\in\{1,\ldots,N\}\end{aligned} \tag{3.11}$$

Given the optimal hyperplane with parameters $\{\boldsymbol{\eta},b\}$ and a new input feature vector $\varphi(\boldsymbol{O})$, then classification can be described as follows:

$$\hat{w}=\text{sign}\Big(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O})+b\Big) \tag{3.12}$$

where $\text{sign}(\cdot)$ is the sign function that extracts the sign (-1 or $+1$) of a real number.

The discussion so far has based on the assumption that the training data is linearly separable. In many real-world applications this assumption is not generally satisfied. In this case the training data are aimed to be separated with a minimal number of errors. To allow training errors, *slack variables*, $\xi_n\geq 0$, are introduced. Then the constraint described in (3.9) is relaxed to $w_n\left(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}_n)+b\right)\geq 1-\xi_n$. This is known as the soft margin SVM constraint. For the data that lie outside the margin edges, $\xi_n=0$. For the misclassified data or the data inside the edges, $\xi_n\geq 0$. The number of training errors is bounded by $\sum_n\xi_n$ [24]. Then, for the non-separable case, the hyperplane is found by minimising the upper bound of the training errors and maximising the margin for the correctly classified

[1] The optimisation problem of minimising $\frac{1}{2}||\boldsymbol{\eta}||$ is difficult to solve because it involves a square root. Alternatively, in SVMs $\frac{1}{2}||\boldsymbol{\eta}||^2$ is minimised.

data [31]:

$$\begin{aligned} \min_{\boldsymbol{\eta},b} & \left\{ \frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\xi_n \right\} \\ \text{s.t.} \quad & w_n\left(\boldsymbol{\eta}^{\mathsf{T}}\varphi(\boldsymbol{O}_n) + b\right) \geq 1 - \xi_n, \quad \forall n \in \{1,\ldots,N\} \\ & \xi_n \geq 0, \quad \forall n \in \{1,\ldots,N\} \end{aligned} \tag{3.13}$$

where C is a constant that is used to trade off between the margin and the training errors. (3.13) is known as the primal form of the SVM optimisation problem. Alternatively, this optimisation problem also can be expressed in a dual form, and more details can be found in [24, 31].

3.1.2.1 *Multi-class SVMs*

Binary classification is considered so far. For multi-class classification tasks, SVMs also can be applied by reducing the multi-class problem into multiple binary classification problems [41, 42, 88, 139]. Rather than decomposing the multi-class problem into multiple binary classification problems, the multi-class SVM [33] has been proposed, which is treated as a single optimization problem.

For multi-class classification, let the parameters for class w be $\boldsymbol{\eta}_w$, the number of classes L and the input feature vector $\varphi(\boldsymbol{O})$, classification then can be expressed as follows[1]:

$$\hat{w} = \arg\max_{w} \left\{ \boldsymbol{\eta}_w^{\mathsf{T}}\varphi(\boldsymbol{O}) \right\} \tag{3.14}$$

In multi-class SVMs, the score for the correct class, such as $\boldsymbol{\eta}_w^{\mathsf{T}}\varphi(\boldsymbol{O})$, is aimed to be greater than the scores for the incorrect classes as much as possible. One such training criterion is expressed as follows [33]:

$$\begin{aligned} \min_{\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_L} & \left\{ \sum_{w=1}^{L}\frac{1}{2}||\boldsymbol{\eta}_w||^2 + C\sum_{n=1}^{N}\xi_n \right\} \\ \text{s.t.} \quad & \boldsymbol{\eta}_{w_n}^{\mathsf{T}}\varphi(\boldsymbol{O}_n) - \boldsymbol{\eta}_w^{\mathsf{T}}\varphi(\boldsymbol{O}_n) \geq \mathcal{L}(w, w_n) - \xi_n, \quad \forall n, w \\ & \xi_n \geq 0, \quad \forall n \in \{1,\ldots,N\} \end{aligned} \tag{3.15}$$

[1] It is worth noting that here the bias is not written out separately, but it can be incorporated in the feature vector, e.g. the features described in equation (3.3).

where L is the total number of classes. $\mathcal{L}(w, w_n)$ is the $1/0$ loss, which is defined as $\mathcal{L}(w, w_n) = 1 - \delta(w, w_n)$, and $\delta(\cdot)$ is the Kronecker delta. Similar to that in multinomial logistic regression (3.6), the parameters $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_L\}$ can be combined to be a single vector:

$$\boldsymbol{\eta} = \left[\boldsymbol{\eta}_1^\mathsf{T}, \ldots, \boldsymbol{\eta}_L^\mathsf{T}\right]^\mathsf{T} \tag{3.16}$$

and let the feature vector be the form described in (3.8) :

$$\boldsymbol{\phi}(\boldsymbol{O}, w) = \begin{bmatrix} \delta(w, 1)\varphi(\boldsymbol{O}) \\ \vdots \\ \delta(w, L)\varphi(\boldsymbol{O}) \end{bmatrix} \tag{3.17}$$

Classification then can be described as follows:

$$\hat{w} = \arg\max_w \left\{\boldsymbol{\eta}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{O}, w)\right\} \tag{3.18}$$

and the training criterion (3.15) becomes:

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \left\{\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\xi_n\right\} \\ &\text{s.t. } \boldsymbol{\eta}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{O}_n, w_n) - \boldsymbol{\eta}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{O}_n, w) \geq \mathcal{L}(w, w_n) - \xi_n, \quad \forall n, w \\ &\quad \xi_n \geq 0, \quad \forall n \in \{1, \ldots, N\} \end{aligned} \tag{3.19}$$

In the discussion so far, only the linear decision boundary (hyperplane) is considered. By applying the *kernel trick* [22], SVMs (or multi-class SVMs) can also be applied to yield a non-linear decision boundary. By using a kernel function, the linear decision boundary is constructed in a transformed feature space. Thus though the decision boundary is a linear hyperplane in the transformed feature space, it might be nonlinear in the original input space. More details on the kernel trick can be found in [22, 31, 33].

3.1.2.2 *Relationships with Logistic Regression*

In the training criterion (3.19) of the multi-class SVM, when substituting the constraints into the minimisation criterion, training can be described as minimising the following objective function:

$$\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\left[\max_w \left\{\mathcal{L}(w, w_n) - \left(\boldsymbol{\eta}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{O}_n, w_n) - \boldsymbol{\eta}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{O}_n, w)\right)\right\}\right]_+ \tag{3.20}$$

where $\left[\cdot\right]_+$ is the hinge loss, which is defined as:

$$\left[f(x)\right]_+ = \max\left\{0, f(x)\right\} = \begin{cases} 0 & \text{when } f(x) < 0 \\ f(x) & \text{when } f(x) \geq 0 \end{cases} \tag{3.21}$$

It is worth noting that the loss function is defined as $\mathcal{L}(w, w_n) = 1 - \delta(w, w_n)$. When $w = w_n$, the function inside the maximisation equals 0. Thus the maximisation function $\max_w\{\cdot\}$ in (3.20) can be expressed as $\max_w\{\cdot\} = \max\left\{0, \max_{w\neq w_n}\{\cdot\}\right\}$. According to the definition of the hinge loss described in (3.21), objective function (3.20) is equivalent to the following expression:

$$\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\left[\max_{w\neq w_n}\left\{\mathcal{L}(w, w_n) - \left(\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w_n) - \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w)\right)\right\}\right]_+ \tag{3.22}$$

These two forms of expressions (3.20) and (3.22) are equivalent, and the later one is used throughout this thesis.

As discussed in [209], large margin training of the logistic regression model is equivalent to a SVM. The margin for logistic regression is defined as the log-posterior ratio between the correct class w_n and the best competing class w. The large margin training criterion can be described as maximising this margin and minimising the loss function $\mathcal{L}(w, w_n)$, which measures the distance between the class w and the correct one w_n. By introducing a Gaussian prior, $\log p(\boldsymbol{\eta}) = \log\mathcal{N}(\boldsymbol{\eta}; \mathbf{0}, C\boldsymbol{I}) = -\frac{1}{2C}||\boldsymbol{\eta}||^2 + \text{Constant}$, with zero mean and scaled identity matrix, large margin training of multinomial logistic regression (3.4) can be described as minimising:

$$\frac{1}{2C}||\boldsymbol{\eta}||^2 + \sum_{n=1}^{N}\left[\max_{w\neq w_n}\left\{\mathcal{L}(w, w_n) - \log\left(\frac{P(w_n|\boldsymbol{O}_n)}{P(w|\boldsymbol{O}_n)}\right)\right\}\right]_+ \tag{3.23}$$

This training criterion has similar form to large margin training of generative models described in (2.23). Substituting the definitions of the multinomial logistic regression (3.4) and (3.7) into criterion (3.23), then large margin training of multinomial logistic regression can be expressed as minimising:

$$\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\left[\max_{w\neq w_n}\left\{\mathcal{L}(w, w_n) - \left(\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w_n) - \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w)\right)\right\}\right]_+ \tag{3.24}$$

This is the training criterion of the multi-class SVM described in (3.22). Therefore, large margin training of multinomial logistic regression models can be interpreted as multi-class

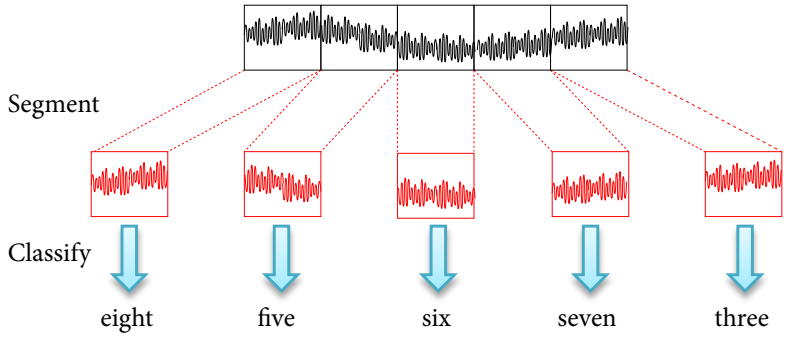


Figure 3.4: *Acoustic code-breaking based on the most likely segmentation [147].*

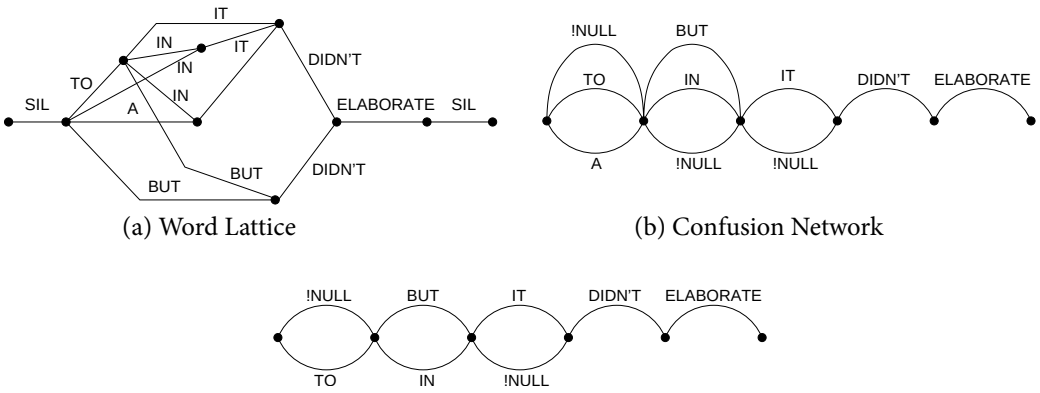
SVMs. It is worth noting that, when a non-zero mean Gaussian prior is used, $\log p(\boldsymbol{\eta}) = \log \mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\mu}_\eta, C\boldsymbol{I}) = -\frac{1}{2C}||\boldsymbol{\eta} - \boldsymbol{\mu}_\eta||^2 + \text{Constant}$, a more general form of large margin training criterion (3.24) can be derived:

$$\frac{1}{2}||\boldsymbol{\eta} - \boldsymbol{\mu}_\eta||^2 + C\sum_{n=1}^{N}\left[\max_{w \neq w_n}\left\{\mathcal{L}(w, w_n) - \left(\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w_n) - \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{O}_n, w)\right)\right\}\right]_+ \tag{3.25}$$

3.1.3 *Acoustic Code-breaking*

In the previous subsections, unstructured discriminative models, such as logistic regression and SVMs, were discussed. In continuous speech recognition (CSR), the inputs are utterances, and the number of possible classes for an utterance is exponential large. Thus, it is impractical to directly model the whole utterance with unstructured discriminative models. One solution to this problem is to use *acoustic code-breaking* [185], where the continuous speech is segmented into segments, and then each segment is treated independently and classified separately. In acoustic code-breaking, the problem of sentence recognition is decomposed into the sub-problems of word (or phone) recognition. These sub-problems then can be addressed directly by using the unstructured discriminative models discussed in the previous subsections.

A number of acoustic code-breaking approaches haven been proposed for CSR [63, 109, 185]. In these approaches, an existing HMM-based speech recogniser is used to yield the 1-best hypothesis or word lattice [179, 200], then classification can be performed based on



(a) Word Lattice

(b) Confusion Network

(c) Pruned Confusion Network

Figure 3.5: *Acoustic code-breaking based on the confusion network [66, 109].*

the given segmentation information. In this subsection, acoustic code-breaking based on the 1-best hypothesis and lattice will be discussed.

Given an input utterance $\boldsymbol{O}$, an existing speech recogniser can be applied to generate the 1-best hypothesis and corresponding segmentation. In this thesis, such segmentation is called the most likely segmentation (given by this recogniser). Given the most likely segmentation, each segment of the utterance can be treated independently, and be classified separately. This type of acoustic code-breaking approach is illustrated in Figure 3.4. In work [63, 196], unstructured classifiers, such as SVMs, were employed in digit classification (where the class labels are zero to nine, oh and silence) under the framework of acoustic code-breaking as illustrated in Figure 3.4.

Alternative to the most likely segmentation, a confusion network can be used in acoustic code-breaking, where binary classification is performed for each word pair. This framework is illustrated in Figure 3.5, and can be summarised in three steps. In the first step, the word lattice is generated by using an existing speech recognition system, and Figure 3.5 (a) gives an example of the word lattice. In the second step, the word lattice is converted to a confusion network, and one example of the confusion network is illustrated in Figure 3.5 (b). In the third step, this confusion network is pruned such that each set of parallel arcs contains at most two as illustrated in Figure 3.5 (c). When the pruned confusion network is produced, binary classifiers can be applied to each confusion word pair. Normally, the binary classifiers are trained only for most frequent confusions which are determined from the training data, and the number of classifiers is limited by the number of available examples in the training

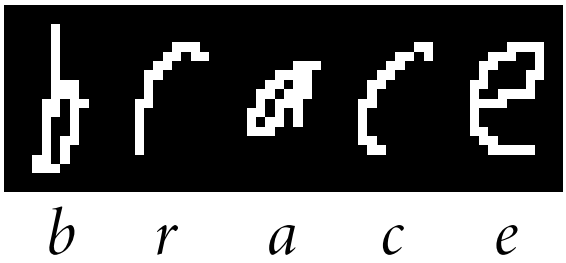
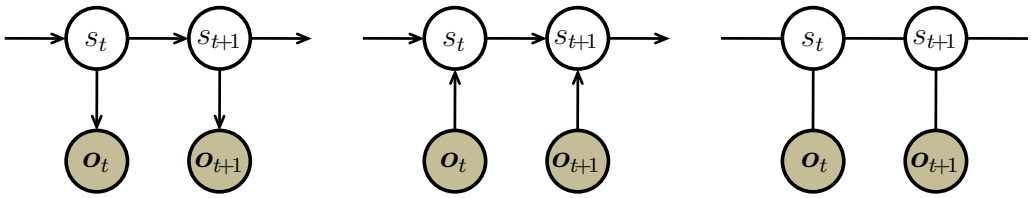


Figure 3.6: *A handwritten word "brace" taken from [173].*

data. This makes acoustic code-breaking a limited approach, especially when the confusion pairs of the test data did not appear in training. Alternative to decompose sentence recognition into word (or phone) recognition by using acoustic code-breaking, structured discriminative models can be directly employed to recognise continuous speech, and this type of model will be discussed in the following section.

3.2 Structured Discriminative Models

In the previous section, unstructured discriminative models were discussed, where the class labels are considered as single (atomic) units and different units distinguish from each other. This type of model cannot be directly applied to speech recognition where the inputs are observation sequence and class labels are sentences, since the number of possible classes for an sentence could be unbounded, e.g. the number of possible classes for a 6-digit string is 10^6. Although the framework of acoustic code-breaking can be applied, unstructured discriminative models only can be used in a very limited way, e.g. fixed segmentations and the limited number of confusion candidates. This motivates interest in structured discriminative models where the structure of the labels is considered. In structured discriminative models, the class label is considered as being comprised of atomic units, and different labels share the common set of units, e.g. different sentences can be considered as being comprised of a sequence of phones from a common set. In structured models, for each atomic unit there is a set of corresponding model parameters, hence the parameters for different classes can be constructed, even these classes did not appear in training. This is similar to HMMs in modelling sentence, where sentence models are constructed by concatenating word (or phone) models. Moreover, the (long range) dependencies within the input sequence can be modelled by structured models, whereas for unstructured models, the input sequence is



(a) HMM (b) MEMM (c) CRF

Figure 3.7: *Graphical representations of the HMM, MEMM and CRF.* (a) *and* (b) *are directed graphs;* (c) *is a undirected graph.*

segmented and different segments are treated independently. Take handwriting recognition for example, for the handwriting illustrated in Figure 3.6, when different letters are treated separately, distinguishing between the second letter 'r' and the fourth letter 'c' in isolation is far from trivial. When the word is treated as a whole, in the context of the surrounding letters this task becomes much less error-prone [173]. Thus, structured discriminative models are better choices in classification of the sequential data with structured labels. In this section, various extensively used structured discriminative models, such as conditional random fields (CRFs) [104] and structured SVMs[1] [209, 210], will be discussed.

3.2.1 Conditional Random Fields

Conditional random fields (CRFs) were original proposed by Lafferty [104] in the machine learning community, and this type of model has been widely used in various fields such as natural language processing (NLP) and automatic speech recognition (ASR) [48]. CRFs are discriminative models, where the conditional probability of the state sequence $S = \{s_1, \ldots, s_T\}$ given the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ is modelled directly[2].

For HMMs, which are generative models, the joint distribution between the state and observation sequences is modelled:

$$p(S, \boldsymbol{O}) = \prod_{t=1}^{T} P(s_t|s_{t-1}) p(\boldsymbol{o}_t|s_t) \tag{3.26}$$

[1] Strictly speaking, structured SVMs are not discriminative models, but discriminant functions which map the inputs to class labels directly.

[2] It is worth noting that only linear-chain CRFs are considered in this work, and the state and observation sequences have the same length.

where $P(s_t|s_{t-1})$ is the state transition probability, and $p(\boldsymbol{o}_t|s_t)$ is the state output probability as discussed in section 2.2. The corresponding graphical model is illustrated in Figure 3.7 (a). It is worth noting that the graphical model in Figure 2.3 is a specification of the HMM (in Figure 3.7 (a)) with GMM state output distribution. Given the joint distribution $p(S, \boldsymbol{O})$, the conditional distribution $P(S|\boldsymbol{O})$ can be obtained through $P(S|\boldsymbol{O}) = p(S, \boldsymbol{O})/p(\boldsymbol{O})$, whereas in *maximum entropy Markov models* (MEMMs) [121], the conditional distribution of the state sequence given observations $P(S|\boldsymbol{O})$ is modelled directly:

$$P(S|\boldsymbol{O}) = \prod_{t=1}^{T} P(s_t|s_{t-1}, \boldsymbol{o}_t) \tag{3.27}$$

where the distribution $P(s_t|s_{t-1}, \boldsymbol{o}_t)$ is modelled by a maximum entropy model (or multinomial logistic regression), with form $P(s_t|s_{t-1}, \boldsymbol{o}_t) \propto \exp\left(\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{o}_t, s_t, s_{t-1})\right)$, as discussed in section 3.1.1. The graphical representation of the MEMM is illustrated in Figure 3.7 (b). Compared with the graphical model of the HMM (in Figure 3.7 (a)), the direction of the arcs from the states to the observations is reversed, that denotes the state distributions now are conditioned on the observations. One major issue of MEMMs is the *label bias problem* [104]. In MEMMs, probability $P(S|\boldsymbol{O})$ is factorised into terms of $P(s_t|s_{t-1}, \boldsymbol{o}_t)$. This means if there are very few states s_t that can follow s_{t-1}, then the role of the observation $\boldsymbol{o}_t$ in distinguishing them is diminished [48]. In the extreme case when there is only one outgoing transition from $s_{t-1}=2$ to s_t (say state 1), then $p(s_t=1|s_{t-1}=2, \boldsymbol{o}_t) = 1$, and consequently $p(s_t \neq 1|s_{t-1}=2, \boldsymbol{o}_t) = 0$. Then the impact of $\boldsymbol{o}_t$ is completely ignored by the model. In MEMMs, the label bias problem arises because for each state the conditional probability $P(s_t|s_{t-1}, \boldsymbol{o}_t)$ is required to be locally normalized to sum to a probability distribution [48].

The label bias problem can be addressed by CRFs [104], where the state sequence S is modelled jointly (given the observation sequence $\boldsymbol{O}$) and a global normalisation is used for the entire conditional distribution $P(S|\boldsymbol{O}, \boldsymbol{\eta})$ (with model parameters $\boldsymbol{\eta}$), which can be expressed as follows:

$$P(S|\boldsymbol{O}, \boldsymbol{\eta}) = \frac{1}{\mathcal{Z}(\boldsymbol{O}, \boldsymbol{\eta})} \exp\left(\boldsymbol{\eta}^{\mathsf{T}}\Phi\big(\boldsymbol{O}, S\big)\right) \tag{3.28}$$

where $\mathcal{Z}(\boldsymbol{O}, \boldsymbol{\eta}) = \sum_S \exp\left(\boldsymbol{\eta}^{\mathsf{T}}\Phi\big(\boldsymbol{O}, S\big)\right)$ is a normalisation term that ensures $P(S|\boldsymbol{O}, \boldsymbol{\eta})$ is a valid probability distribution. $\Phi\big(\boldsymbol{O}, S\big)$[1] is the joint feature vector, which can be ex-

[1] In this thesis, the joint feature vector, that characterises the dependence between the observations and the label sequence (where the label is a sequence rather than a single unit), is denoted as $\Phi(\cdot)$.

pressed as the form consisting of acoustic and language features [209]:

$$\Phi(\boldsymbol{O}, S) = \begin{bmatrix} \phi_{\text{ac}}(\boldsymbol{O}, S) \\ \phi_{\text{lg}}(S) \end{bmatrix} = \sum_{t=1}^{T} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{o}_t, s_t) \\ \boldsymbol{\phi}(s_t, s_{t-1}) \end{bmatrix} \tag{3.29}$$

where $\boldsymbol{\phi}(\boldsymbol{o}_t, s_t)$ and $\boldsymbol{\phi}(s_t, s_{t-1})$ are the acoustic and language feature vectors for frame $\boldsymbol{o}_t$. One simple form of the features is defined as the sufficient statistics with respect to the HMMs:

$$\boldsymbol{\phi}(\boldsymbol{o}_t, s_t) = \begin{bmatrix} \vdots \\ \delta(s_t = i) \\ \delta(s_t = i)\boldsymbol{o}_t \\ \delta(s_t = i)\text{diag}(\boldsymbol{o}_t \boldsymbol{o}_t^{\mathsf{T}}) \\ \vdots \end{bmatrix}, \qquad \forall i \tag{3.30}$$

$$\boldsymbol{\phi}(s_t, s_{t-1}) = \begin{bmatrix} \vdots \\ \delta(s_t = i) \\ \delta(s_t = i, s_{t-1} = j) \\ \vdots \end{bmatrix}, \qquad \forall i, j \tag{3.31}$$

where $\delta(\cdot)$ is the Kronecker delta. Thus, the position of $\boldsymbol{o}_t$ and $\text{diag}(\boldsymbol{o}_t \boldsymbol{o}_t^{\mathsf{T}})$ in the feature vector $\boldsymbol{\phi}(\boldsymbol{o}_t, s_t)$ depends on the state label i. The features $\boldsymbol{\phi}(\boldsymbol{o}_t, s_t)$ are associated with the state output probabilities, and the features $\boldsymbol{\phi}(s_t, s_{t-1})$ are associated with the HMM transition probabilities. Alternative to the sufficient statistics with respect to the HMMs, log-likelihoods and their partial derivatives [106, 148] also can be used as the features. The form of the features based on log-likelihoods will be discussed in section 3.5.

3.2.2 *Hidden Conditional Random Fields*

In CRFs, the class labels are sate sequences, and the conditional distribution of the state sequence given the observations $P(S|\boldsymbol{O})$ is modelled. However, in some applications, the class labels are not directly linked to the observations, and the underlying hidden variables of the class needs to be considered. Speech recognition is a typical example, where the conditional distribution of the word (or sub-word) sequence given observations $P(W|\boldsymbol{O})$ is modelled, and the state sequence is treated as hidden variables, which are marginalised out in calculating the conditional distribution $P(W|\boldsymbol{O})$. Thus, *hidden conditional random*

fields (HCRFs) [79, 145] were proposed to generalise CRFs to handle the hidden variables underlying the class labels. In HCRFs, the conditional distribution of the word (or sub-word) sequence $W = \{w_1, \ldots, w_I\}$ given the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ can be expressed as follows:

$$P(W|\boldsymbol{O}, \boldsymbol{\eta}) = \frac{1}{\mathcal{Z}(\boldsymbol{O}, \boldsymbol{\eta})} \sum_{S} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}, W, S)\right) \tag{3.32}$$

where $S = \{s_1, \ldots, s_T\}$ is the state sequence corresponding to the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$, and the sum is over all possible state sequences S associated with label W. $\boldsymbol{\eta}$ are model parameters and $\mathcal{Z}(\boldsymbol{O}, \boldsymbol{\eta}) = \sum_W \sum_S \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}, W, S)\right)$ is the normalisation term. Compared with CRFs described in (3.28), in HCRFs the hidden variables are marginalised out, and optimisation is no longer convex, so a global optimum on the training set is not guaranteed [48]. In (3.32), the form of the joint feature vector $\Phi(\boldsymbol{O}, W, S)$ has a similar form to that in CRFs (3.29):

$$\Phi(\boldsymbol{O}, W, S) = \begin{bmatrix} \boldsymbol{\phi}_{\text{ac}}(\boldsymbol{O}, W, S) \\ \boldsymbol{\phi}_{\text{lg}}(W, S) \end{bmatrix} = \sum_{t=1}^{T} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{o}_t, s_t) \\ \boldsymbol{\phi}(s_t, s_{t-1}) \end{bmatrix} \tag{3.33}$$

where S is a state sequence associated with the word (or sub-word) sequence W. $\boldsymbol{\phi}(\boldsymbol{o}_t, s_t)$ and $\boldsymbol{\phi}(s_t, s_{t-1})$ are acoustic and language feature vectors described in (3.30) and (3.31).

In CRFs and HCRFs, the model structure is embodied in the formation of joint feature vector $\Phi(\cdot)$. The model parameters and joint feature vectors for a sentence can be constructed by combining the local parameters and features associated with the states and words (or sub-words) [209, 210]. This is similar to the HMM for an sentence, which is formed by concatenating the HMM word (or sub-word) models.

3.2.3 Segmental Conditional Random Fields

Segmental conditional random fields (SCRFs) are the extension of CRFs. SCRFs relax the Markov assumption from the frame level to the segment level. This means the Markov assumption is not enforced on the observations within the segment, that allows to capture long-span dependencies [48]. This type of model is also known as the *semi-Markov CRF* [154]. Let $\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_I\}$ be an segmentation corresponding to observations $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \ldots, \boldsymbol{O}_{(I)}\}$ with label $W = \{w_1, \ldots, w_I\}$, where $\boldsymbol{O}_{(i)}$ is one segment of observations, and ρ_i gives segmentation information for the word (or sub-word) label w_i, e.g.

the index of the frames associated with label w_i. The conditional distribution modelled by SCRFs can be expressed as:

$$P(W|\boldsymbol{O},\boldsymbol{\eta}) = \frac{1}{\mathcal{Z}(\boldsymbol{O},\boldsymbol{\eta})} \sum_{\boldsymbol{\rho}} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}, W, \boldsymbol{\rho})\right) \tag{3.34}$$

This type of model is also known as the *conditional augmented model* (CAug) [106]. In the model definition (3.34), the sum is over all possible segmentations $\boldsymbol{\rho}$ associated with label W. $\mathcal{Z}(\boldsymbol{O},\boldsymbol{\eta}) = \sum_W \sum_{\boldsymbol{\rho}} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}, W, \boldsymbol{\rho})\right)$ is the normalisation term same as that in CRFs and HCRFs. The joint feature vector $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$ allows the observations within a segment to be related, rather than making conditional independence assumption on the observations in CRFs and HCRFs [209]:

$$\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) = \begin{bmatrix} \boldsymbol{\phi}_{\text{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) \end{bmatrix} = \sum_{i=1}^{|\boldsymbol{\rho}|} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{O}_{(i)}, w_i) \\ \boldsymbol{\phi}(w_i, w_{i-1}) \end{bmatrix} \tag{3.35}$$

where $|\boldsymbol{\rho}|$ is the number of segments. $\boldsymbol{\phi}(\boldsymbol{O}_{(i)}, w_i)$ and $\boldsymbol{\phi}(w_i, w_{i-1})$ are the segment level acoustic and language feature vectors. Let $\{v_1, \ldots, v_L\}$ denote all unique sub-sentence units (words or sub-words) in the dictionary, the acoustic feature vector can be expressed as:

$$\boldsymbol{\phi}(\boldsymbol{O}_{(i)}, w_i) = \begin{bmatrix} \vdots \\ \delta(w_i = v_l)\varphi(\boldsymbol{O}_{(i)}) \\ \vdots \end{bmatrix}, \qquad \forall l \tag{3.36}$$

where $\varphi(\cdot)$ is the feature vector for a segment, which maps the observations with various length to a vector with fixed dimension. One possible expression of $\varphi(\cdot)$ is comprised of likelihoods:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p(\boldsymbol{O}_{(i)}|v_1) \\ \vdots \\ \log p(\boldsymbol{O}_{(i)}|v_L) \end{bmatrix} \tag{3.37}$$

The language feature vector $\boldsymbol{\phi}(w_i, w_{i-1})$ is related to unigram and bigram language models, and might be described as [209]:

$$\boldsymbol{\phi}(w_i, w_{i-1}) = \begin{bmatrix} \vdots \\ \delta(w_i = v) \\ \delta(w_i = v, w_{i-1} = v') \\ \vdots \end{bmatrix}, \qquad \forall v, v' \tag{3.38}$$

where v and v' are the possible words (or sub-words) in the dictionary. The graphical representation of a HCRF is illustrated in Figure 3.8.

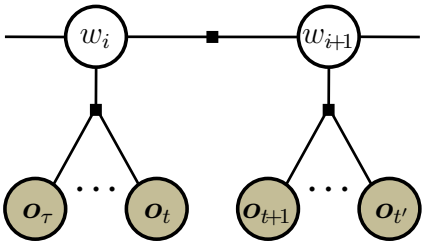


Figure 3.8: *The factor graph representation of a segmental CRF.*

3.2.4 Structured Log-linear Models

In the precious subsection, segmental CRFs (or conditional augmented models) were introduced. In this type of model, the sum over all possible segmentations leads to a non-convex optimisation problem and inefficiency in training. Analogous to Viterbi decoding [46, 187], where the likelihood is calculated by finding the most likely state sequence, here the most likely segmentation $\boldsymbol{\rho}$ is used instead of summing over all possible segmentations [147]. Then segmental CRFs described in (3.34) can be approximated as a *log-linear model*:

$$P(W|\boldsymbol{O},\boldsymbol{\eta}) \approx \frac{1}{\mathcal{Z}(\boldsymbol{O},\boldsymbol{\eta})} \exp\left(\boldsymbol{\eta}^{\mathsf{T}}\Phi\big(\boldsymbol{O},W,\boldsymbol{\rho}\big)\right) \tag{3.39}$$

The most likely segmentation $\boldsymbol{\rho}$ can be obtained from the generative model (HMM). Given the HMM with parameters $\boldsymbol{\lambda}$, the most likely segmentation $\boldsymbol{\rho}$ can be estimated by maximising [54]:

$$\boldsymbol{\rho}_\lambda = \arg\max_{\boldsymbol{\rho}} P(\boldsymbol{\rho}) p(\boldsymbol{O}|\boldsymbol{\lambda},\boldsymbol{\rho}) \tag{3.40}$$

where $p(\boldsymbol{O}|\boldsymbol{\lambda},\boldsymbol{\rho})$ is the likelihood given by the HMM. In this work, the probabilities of choosing different segments are supposed to be equal, namely $P(\boldsymbol{\rho})$ is a uniform distribution. Alternatively, the optimal segmentation obtained from discriminative models is discussed in [210]. In this work, the best segmentation from discriminative model is only considered in classification, where the best segmentation aims to maximise the conditional probability described in (3.39), i.e. $\max_{\boldsymbol{\rho}} \boldsymbol{\eta}^{\mathsf{T}}\Phi\big(\boldsymbol{O},W,\boldsymbol{\rho}\big)$ for the numerator term. Then classification with this log-linear model can be described as follows [210]:

$$\hat{W} = \arg\max_{W} \left\{ \max_{\boldsymbol{\rho}} \boldsymbol{\eta}^{\mathsf{T}}\Phi\big(\boldsymbol{O},W,\boldsymbol{\rho}\big) \right\} \tag{3.41}$$

This yields both the optimal word sequence and segmentation.

It is worth noting that multinomial logistic regression discussed in section 3.1.1 and CRFs in section 3.2.1 also belong to the framework of log-linear models, but with different definitions of the feature function $\Phi(\cdot)$.

3.2.5 Structured SVMs

Structured SVMs are generalisations of SVMs to handle the sequential data with structured labels. Similar to the SVM, the structured SVM is a discriminant function, where the (sequential) input is mapped to a (sequential) label directly by maximising the discriminant function:

$$\hat{W} = \arg\max_{W} \left\{ \boldsymbol{\eta}^{\mathsf{T}} \Phi\big(\boldsymbol{O}, W, \boldsymbol{\rho}\big) \right\} \tag{3.42}$$

where $\boldsymbol{\eta}$ are the model parameters. $\Phi(\cdot)$ is the joint feature vector. One example is given in (3.35) and various forms of the joint features will be discussed in detail in section 3.5. In speech recognition, the segmentation $\boldsymbol{\rho}$ is seldom known. Thus, in classification the optimal segmentation needs to be estimated with the class label:

$$\big(\hat{W}, \hat{\boldsymbol{\rho}}\big) = \arg\max_{W, \boldsymbol{\rho}} \left\{ \boldsymbol{\eta}^{\mathsf{T}} \Phi\big(\boldsymbol{O}, W, \boldsymbol{\rho}\big) \right\} \tag{3.43}$$

This is equivalent to classification with log-linear models as described in (3.41).

Given the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$ comprised of utterance and label pairs, and the corresponding most likely segmentations $\{\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_N\}$ from HMMs as described in (3.40). The training criterion of the structured SVM can be expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \left\{ \frac{1}{2} ||\boldsymbol{\eta}||^2 + C \sum_{n=1}^{N} \xi_n \right\}, \qquad \forall n, \xi_n \geq 0 \\ &\text{s.t.} \quad \forall n, (W, \boldsymbol{\rho}) \neq (W_n, \boldsymbol{\rho}_n) : \\ &\qquad \boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) - \boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \geq \mathcal{L}(W, W_n) - \xi_n, \\ &\qquad \xi_n \geq 0 \end{aligned} \tag{3.44}$$

where $(W, \boldsymbol{\rho})$ is any possible label and segmentation pair, and it is different to the reference and most likely segmentation pair $(W_n, \boldsymbol{\rho}_n)$. $\mathcal{L}(W, W_n)$ is the loss function between label

W and reference W_n. Similar to the SVM, the training criterion (3.44) can be written as minimising follows:

$$\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\mathcal{L}(W,W_n)- \Big(\boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\boldsymbol{\rho}_n) - \boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\boldsymbol{\rho})\Big)\Big\}\Bigg]_+ \tag{3.45}$$

where $\left[\cdot\right]_+$ is the hinge loss defined in (3.21). For each training instance, the best competing label and segmentation pair $(W,\boldsymbol{\rho})$ is found over all possible labels and segmentations[1] except the reference with the corresponding segmentation $(W_n,\boldsymbol{\rho}_n)$.

3.2.5.1 *Relationships with Log-linear Models*

Structured SVMs are closely related to the log-linear models discussed in section 3.2.4. Analogous to the relationship between multinomial logistic regression and the SVM (discussed in section 3.1.2.2), the structured SVM can be interpreted as large margin training of the log-linear model described in (3.39) [210]. When the margin is defined as the log-posterior ratio of the log-linear models between the reference W_n and the best competing label W, the large margin training criterion can be described as maximising this margin and minimising the loss function $\mathcal{L}(w,w_n)$. By introducing a prior $p(\boldsymbol{\eta})$, large margin training of the log-linear model described in (3.39) can be described as minimising [210]:

$$-\log p(\boldsymbol{\eta}) + \sum_{n=1}^{N}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\mathcal{L}(W,W_n) - \log\Big(\frac{P(W_n|\boldsymbol{O}_n,\boldsymbol{\eta})}{P(W|\boldsymbol{O}_n,\boldsymbol{\eta})}\Big)\Big\}\Bigg]_+ \tag{3.46}$$

When the prior $p(\boldsymbol{\eta})$ is a Gaussian distribution, $p(\boldsymbol{\eta}) = \mathcal{N}(\mathbf{0},C\boldsymbol{I}) \propto \exp(-\frac{1}{2C}||\boldsymbol{\eta}||^2)$, with zero mean and scaled identity matrix, and substituting the definition of the log-linear model (3.39) into the large margin training criterion (3.46), the denominator terms of the log-linear models can be cancelled out. Then, the training criterion (3.46) can be further written as minimising:

$$\frac{1}{2}||\boldsymbol{\eta}||^2 + C\sum_{n=1}^{N}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\mathcal{L}(W,W_n)- \Big(\boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\boldsymbol{\rho}_n) - \boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\boldsymbol{\rho})\Big)\Big\}\Bigg]_+ \tag{3.47}$$

[1] These possible labels and segmentations can be obtained from a denominator lattice [147, 209].

This is the training criterion of the structured SVM described in (3.45). In some applications, the prior knowledge on the model parameters $\boldsymbol{\eta}$ is available, then a non-zero mean Gaussian distribution can be introduced, e.g. $p(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\mu}, C\boldsymbol{I}) \propto \exp(-\frac{1}{2C}||\boldsymbol{\eta} - \boldsymbol{\mu}||^2)$. $\boldsymbol{\mu}$ is the available model parameters, e.g. the parameters of the log-linear model trained with the conditional maximum likelihood criterion. With this non-zero mean Gaussian distribution, large margin training of the log-linear model can be described as follows:

$$\frac{1}{2}||\boldsymbol{\eta} - \boldsymbol{\mu}||^2 + C\sum_{n=1}^{N}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\mathcal{L}(W, W_n) - \Big(\boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) - \boldsymbol{\eta}^{\mathsf{T}}\Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho})\Big)\Big\}\Bigg]_+ \tag{3.48}$$

3.3 Training Criteria for Discriminative Models

Different structured discriminative models were discussed in the precious section. Similar to generative models, various training criteria can be employed in training these discriminative models. In this section, the commonly used training criteria for discriminative models will be discussed. Let the training data for discriminative models be $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, which consist of utterance and reference pairs. Given these training data, the model parameters can be estimated by maximising (or minimising) the training criteria, and these criteria will be discussed in the following subsections.

3.3.1 Conditional Maximum Likelihood (CML)

Similar to maximum likelihood estimation for generative models, *conditional maximum likelihood* (CML) training is extensively used in parameter estimation of discriminative models. In CML estimation, given the training data $\mathcal{D}$, the conditional likelihood of the model parameters is maximised. The objective function of CML estimation can be expressed as follows:

$$\mathcal{F}_{\text{CML}}(\boldsymbol{\eta}) = \sum_{n=1}^{N}\log p(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}) \tag{3.49}$$

It is worth noting that this training criterion (3.49) has the same form as the maximum mutual information (MMI) training criterion (2.16) for the generative models. The difference is

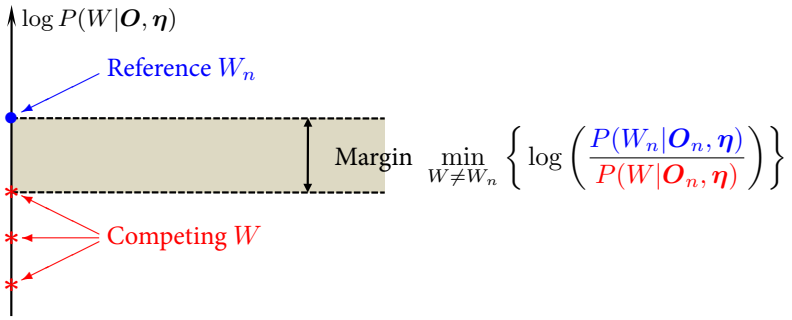


Figure 3.9: *The margin definition for discriminative models.*

that in the MMI criterion the conditional distributions are written in the form of consisting of likelihoods (from generative models) through Bayes' rule as described in (2.17).

In CML training, the objective function (3.49) is to be maximised. Normally, the iterative training approaches, such as gradient based algorithms, are used in parameter estimation. For CRFs and log-linear models, the objective function is convex, hence a global optimum can be found in parameter estimation; whereas, for hidden CRFs and segmental CRFs, due to the sum over the latent variables, the objective function is no longer convex, and a global optimum is not guaranteed [48].

3.3.2 *Minimum Bayes Risk (MBR)*

In discriminative training of generative models, *minimum Bayes risk* (MBR), which minimises the expected loss on the training data, is one of the most extensively used training criteria. Similarly, this type of training criterion also can be applied to discriminative models. In MBR training, the objective function to be minimised can be expressed as follows:

$$\mathcal{F}_{\text{MBR}}(\boldsymbol{\eta}) = \sum_{n=1}^{N} \sum_{W} P(W|\boldsymbol{O}_n,\boldsymbol{\eta}) \mathcal{L}(W, W_n) \tag{3.50}$$

where W denotes all possible word sequences. $\mathcal{L}(W, W_n)$ is the loss function, that measures how different the word sequence W and the reference W_n are. Analogous to MBR training of generative models discussed in section 2.3.4, there are a number of definitions for the loss function, and these definitions lead to different meaningful training criteria, such as the *minimum word error* (MWE) and *minimum phone error* (MPE) criteria.

3.3.3 Large Margin Training

In large margin training, the margin is defined as the log-posterior ratio of the structured discriminative models between the reference W_n and the best competing label W (this margin definition is illustrated in Figure 3.9). Then the training criterion can be described as minimising the inverse of this margin and the loss function $\mathcal{L}(W, W_n)$ [65]:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\eta}) = \sum_{n=1}^{N} \left[\max_{W \neq W_n} \left\{ \mathcal{L}(W, W_n) - \log \left(\frac{P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta})}{P(W|\boldsymbol{O}_n, \boldsymbol{\eta})} \right) \right\} \right]_+ \tag{3.51}$$

where $\left[\cdot\right]_+$ is the hinge loss defined in (3.21). The maximum is found over all possible labels W except the correct one W_n. This criterion (3.51) has the same form as the large margin training criterion (2.23) for generative models. The difference is that for generative models the conditional distributions $P(W|\boldsymbol{O})$ are written in the form of consisting of likelihoods through Bayes' rule as described in equation (2.17).

For segmental CRFs (also apply to hidden CRFs), by substituting the model definition (3.34) into the large margin training criterion (3.51), the denominator terms of the discriminative models can be cancelled out. Then criterion (3.51) can be further written as follows:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\eta}) = \sum_{n=1}^{N} \Bigg[\overbrace{\max_{W \neq W_n} \left\{ \mathcal{L}(W, W_n) + \log \sum_{\boldsymbol{\rho}} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \right) \right\}}^{\text{convex}} \underbrace{- \log \sum_{\boldsymbol{\rho}} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}) \right)}_{\text{concave}} \Bigg]_+ \tag{3.52}$$

This optimisation problem can be solved by using the *concave-convex procedure* (CCCP) [204] and cutting plane algorithm [96], but efficiency might be an issue. First, the sum over all possible segments leads to inefficiency in training and decoding. Moreover, this objective function is not convex, although it is comprised of concave and convex functions. For Log-linear models (and CRFs), this large margin training criterion (3.51) (with a Gaussian prior) is equivalent to the training criterion of the structured SVM as discussed in section 3.2.5.1.

3.4 Adaptation for Discriminative Models

As discussed in section 2.4, in speech recognition the mismatch between the training and test data is a common issue. In order to address this mismatch problem, a range of adapta-

tion approaches, such as maximum a posteriori (MAP) and linear transform based schemes, have been proposed. Analogously, for discriminative models, adaptation also can be applied, and the adaptation approaches can be split into three broad categories: general adaptation, linear transform based approaches and feature adaptation [65]. For generative models, the majority of adaptation approaches are based on maximum likelihood (ML), whereas for discriminative models, adaptation is usually based on conditional maximum likelihood (CML).

In general adaptation approaches, there is no assumption made on the nature of the features in the model. Maximum a posteriori (MAP) [27, 170] is a typical example. Similar to adaptation for generative models discussed in section 2.4.1, MAP adaptation for discriminative models can be described as maximising the follows:

$$\begin{aligned}\mathcal{F}_{\text{MAP}}(\boldsymbol{\eta}) &= \log p(\boldsymbol{\eta}) + \mathcal{F}_{\text{CML}}(\boldsymbol{\eta}) \\ &= \log p(\boldsymbol{\eta}) + \sum_{n=1}^{N} \log p(W_n|\boldsymbol{O}_n, \boldsymbol{\eta})\end{aligned} \tag{3.53}$$

where $\mathcal{F}_{\text{CML}}(\boldsymbol{\eta})$ is the CML criterion described in (3.49), and $p(\boldsymbol{\eta})$ is the prior distribution over parameters $\boldsymbol{\eta}$. This prior distribution is often chosen to be a Gaussian distribution. When the Gaussian prior has the non-zero mean and scaled identity covariance matrix, namely $p(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\mu}, C\boldsymbol{I}) \propto \exp\big(-\frac{1}{2C}||\boldsymbol{\eta} - \boldsymbol{\mu}||^2\big)$, MAP adaptation described in (3.53) can be expressed as the CML criterion with a regularisation term:

$$\mathcal{F}'_{\text{MAP}}(\boldsymbol{\eta}) = -\frac{1}{2C}||\boldsymbol{\eta} - \boldsymbol{\mu}||^2 + \mathcal{F}_{\text{CML}}(\boldsymbol{\eta}) \tag{3.54}$$

It is worth noting that the Gaussian prior also can be used in large margin training (3.51). As discussed in section 3.2.5.1, when the discriminative models are log-linear models, large margin training with a Gaussian prior is equivalent to the training criterion of structured SVMs.

Although the general approaches can be applied in discriminative models, they do not take advantage of any structure in the features [65]. Alternative approaches, such as linear transform based schemes, have been proposed in [117, 169]. These approaches make use of linear transforms similar to that for HMMs discussed in section 2.4.2. The third form of adaptation is related to feature compensation schemes used in generative models [65]. In this approach, the features are modified to be independent of the speaker or environment,

rather than adapting the model parameters. These adapted features can be generated based on generative models, and various adaptation techniques (some of these were discussed in section 2.4) can used for these generative models. The features based on generative models will be discussed in detail in the following section.

3.5 Features for Discriminative Models

Various structured discriminative models have been introduced in section 3.2. In this section, the possible forms of the feature vector for segmental CRFs, log-linear models and structured SVMs will be discussed in detail. In general, the features for these discriminative models can be expressed as a form consisting of the acoustic and language features[1] [209]:

$$\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) = \begin{bmatrix} \phi_{\text{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \phi_{\text{lg}}(W, \boldsymbol{\rho}) \end{bmatrix} \tag{3.55}$$

where $\Phi(\cdot)$ is the joint feature function, which characterise the dependencies between the observations and label sequence, and the corresponding feature vector $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$ is known as the joint feature vector. $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ are the observations (or utterance), W is the class label (or sentence) which is a word (or sub-word) sequence, and $\boldsymbol{\rho}$ is the segmentation. Given the segmentation $\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_I\}$ where I is the number of segments and ρ_i gives information for the ith segment, e.g. the frames index associated with the ith segment, the observations and corresponding label sequence can be described as $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \ldots, \boldsymbol{O}_{(I)}\}$ and $W = \{w_1, \ldots, w_I\}$, where $\boldsymbol{O}_{(i)}$ is a segment and w_i is a word or sub-word unit. This type of notation is used throughout this thesis. In the feature definition (3.55), $\phi_{\text{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho})$ are the acoustic features which are related to the observation statistics, and $\phi_{\text{lg}}(W, \boldsymbol{\rho})$ are the language features that relate to pronunciation probabilities and word statistics [65, 209]. Normally, the acoustic features for speech recognition can be divided into to two categories, namely the frame level and segment level features. The acoustic features can be expressed in a general form of summing over the features for all segments:

$$\phi_{\text{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) = \sum_{i=1}^{I} \phi(\boldsymbol{O}_{(i)}, w_i, \rho_i) \tag{3.56}$$

[1] In some literatures, the language features are also called supra-segmental features [65, 147].

where $\boldsymbol{\phi}(\boldsymbol{O}_{(i)}, w_i, \rho_i)$ is the feature vector for one segment. In the following subsections, the forms of the acoustic and language features will be discussed in detail.

3.5.1 Frame Level Features

When using frame level features, in this thesis another type of hidden information is considered as being associated with the segmentation $\boldsymbol{\rho}$, i.e. the state index $S = \{s_1, \ldots, s_T\}$ corresponding to the observations $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$. Let L be the total number of unique states, the frame level features for a segment then can be expressed as follows [65]:

$$\boldsymbol{\phi}(\boldsymbol{O}_{(i)}, w_i, \rho_i) = \sum_{t \in \{\rho_i\}} \boldsymbol{\phi}(\boldsymbol{o}_t, s_t) = \sum_{t \in \{\rho_i\}} \begin{bmatrix} \delta(s_t, 1)\varphi(\boldsymbol{o}_t) \\ \vdots \\ \delta(s_t, L)\varphi(\boldsymbol{o}_t) \end{bmatrix} \tag{3.57}$$

where $\{\rho_i\}$ denotes the index of the frames associated with the ith segment, and $\delta(\cdot)$ is the Kronecker delta. It is worth noting that since the segmentation information is given, s_t (where $t \in \{\rho_i\}$) only denotes the state associated with w_i, and this also applies to the following discussion. In (3.57) $\varphi(\boldsymbol{o}_t)$ is the feature (vector) for observation $\boldsymbol{o}_t$, and various definitions of $\varphi(\boldsymbol{o}_t)$ will be discussed in the following subsections. By substituting this form of frame level features (3.57) in, the acoustic features defined in (3.56) can be further expressed as follows:

$$\boldsymbol{\phi}_{\text{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) = \sum_{t=1}^{T} \boldsymbol{\phi}(\boldsymbol{o}_t, s_t) = \sum_{t=1}^{T} \begin{bmatrix} \delta(s_t, 1)\varphi(\boldsymbol{o}_t) \\ \vdots \\ \delta(s_t, L)\varphi(\boldsymbol{o}_t) \end{bmatrix} \tag{3.58}$$

3.5.1.1 Gaussian Statistic and Log-likelihood Features

A simple form of the features $\varphi(\boldsymbol{o}_t)$ is the one used in HCRFs [79], which consists of the Gaussian sufficient statistics of observations:

$$\varphi(\boldsymbol{o}_t) = \begin{bmatrix} 1 \\ \boldsymbol{o}_t \\ \text{diag}(\boldsymbol{o}_t \boldsymbol{o}_t^{\mathsf{T}}) \end{bmatrix} \tag{3.59}$$

when the parameters corresponding to this feature vector are:

$$\boldsymbol{\eta}_t = \begin{bmatrix} -\frac{1}{2}\left(k \log(2\pi) + \log|\boldsymbol{\Sigma}| + \log\left(\boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right)\right) \\ \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \\ -\frac{1}{2}\text{diag}\left(\boldsymbol{\Sigma}^{-1}\right) \end{bmatrix} \tag{3.60}$$

where k is the dimension of the observation $\boldsymbol{o}_t$. Given this parameter definition (3.60), the dot product of the feature vector and parameters is the log-likelihood for a Gaussian distribution[1]:

$$\boldsymbol{\eta}_t^{\mathsf{T}} \varphi(\boldsymbol{o}_t) = \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.61}$$

where $\boldsymbol{\mu}$ is the mean of the Gaussian distribution, and $\boldsymbol{\Sigma}$ is a diagonal covariance matrix.

By using this form of parameters (3.60), state output distributions of the HMM can be retrieved. This is the reason why CRFs and HCRFs can retrieve the HMM probabilities [79], which means the HMM baseline can be achieved by setting the parameters in a fashion described in (3.60). This also motivates the use of log-likelihoods as features, where the dimensionality of the features can be dramatically reduced compared with the Gaussian statistic features. The simplest form of the log-likelihood features for an frame can be expressed as follows:

$$\varphi(\boldsymbol{o}_t) = \left[\begin{array}{c} \log p(\boldsymbol{o}_t|s_t) \end{array} \right] \tag{3.62}$$

where $p(\boldsymbol{o}_t|s_t)$ is the state output distribution. Thus, it is straightforward that the state output distribution can be retrieved by setting the corresponding parameter to be 1. In additional to log-likelihoods, features can be based on classifiers which provide information about the discrimination between word (or sub-word) classes [65]. Multilayer perceptron (MLP) [15] is one such example, which provides the posterior probability of phone units.

In terms of log-likelihood features, when the log-likelihoods are given by the DNN-HMM hybrid system, where the likelihoods are obtained according to Bayes' rule as described in equation (2.9): $p(\boldsymbol{o}_t|s_t) \propto P(s_t|\boldsymbol{o}_t)/P(s_t)$ [2]. The softmax activation function is often used for the output layer of the DNN. Let the outputs of the last hidden layer be $\boldsymbol{h}$, and $\boldsymbol{A}$ be the weights for the output layer, then the dot production of the feature $\varphi(\boldsymbol{o}_t)$ described in (3.62) and the corresponding parameter η_t can be expressed as:

$$\eta_t \varphi(\boldsymbol{o}_t) = \eta_t A_{s_t}^{\mathsf{T}} \boldsymbol{h} + \eta_t f(s_t) \tag{3.63}$$

where A_{s_t} is the row of $\boldsymbol{A}$ that is associated with state s_t, and $f(s_t)$ is a state dependent function, which is the logarithm of the normalisation term that ensures $p(\boldsymbol{o}_t|s_t)$ be a valid

[1] Without loss of generality, a single Gaussian distribution is considered here.

[2] Since $p(\boldsymbol{o}_t)$ is not a function of the state s_t, and it has a equal impact on the DNN outputs, hence it is left out in the analysis.

distribution. According to this expression, it is interesting to note that the features extracted from the hybrid system can be viewed as being transformed from the hidden layer outputs [38]. As described in (3.63), discriminative models introduce additional parameters in the feature transformation, thus optimal transforms might be learnt with an appropriate training criterion.

3.5.1.2 *Features Based on Multiple Systems*

In general, different systems have various characteristics, make different errors, and are expected to provide complementary advantages. Thus, in speech recognition, system combination approaches are of growing interest, and state-of-the-art speech recognisers typically utilise multiple systems to make ensemble decisions. Analogously, features can be extracted from multiple systems. In this work, only the log-likelihood based features are considered. The features for an frame based on D different systems can be described as:

$$\varphi(\boldsymbol{o}_t) = \begin{bmatrix} \log p_1(\boldsymbol{o}_t|s_t) \\ \vdots \\ \log p_D(\boldsymbol{o}_t|s_t) \end{bmatrix} \tag{3.64}$$

where $p_d(\boldsymbol{o}_t|s_t)$ is the log-likelihood given by the dth system. According to the feature definition (3.64), in generating the features, the likelihoods are given by the output distributions of the same state associated different systems. Thus, these different systems share the same HMM topology as that in the joint decoding system discussed in section 2.2.

When using frame level features based on multiple systems, by substituting the feature definitions (3.64), (3.58) and (3.55) in, decoding with log-linear models described in (3.41) can be further expressed as follows[1]:

$$\hat{W} = \arg\max_{W} \left\{ \max_{S} \left\{ \boldsymbol{\eta}^{\text{lg}} \boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) + \sum_{t=1}^{T} \sum_{j=1}^{L} \delta(s_t, j) \sum_{d=1}^{D} \eta_{j,d} \log p_d(\boldsymbol{o}_t|s_t) \right\} \right\} \tag{3.65}$$

where t is the index for frames, j is the jth unique state, and d is the index for systems. $\boldsymbol{\eta}^{\text{lg}}$ and $\eta_{j,d}$ are the parameters corresponding to features $\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho})$[2] and $\log p_d(\boldsymbol{o}_t|s_t)$[3],

[1] As discussed at the beginning of section 3.5.1, the hidden states S are considered as being associated with the segmentation $\boldsymbol{\rho}$. When using frame-level features, maximisation yields the best word and state sequences.

[2] The language model features $\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho})$ are the elements of the joint feature vector described in (3.55).

[3] $\log p_d(\boldsymbol{o}_t|s_t)$ is the dth element of the feature vector described in (3.64).

and they are elements of the whole model parameters $\boldsymbol{\eta}$. For the tth frame, in decoding as described in (3.65), the acoustic score computed for frame $\boldsymbol{o}_t$ can be described as:

$$\mathcal{L}(\boldsymbol{o}_t|s_t=j) = \sum_{d=1}^{D} \eta_{j,d} \log p_d(\boldsymbol{o}_t|s_t=j) \tag{3.66}$$

This is the same as the combined score used in joint decoding described in (2.10), but with more general state dependent combination weights $\boldsymbol{\eta}_j^{\mathsf{T}} = \left[\eta_{j,1}, \ldots, \eta_{j,D}\right]$, which vary with state j[1]. When the language score $\boldsymbol{\eta}^{\text{lg}}\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho})$ is comprised of the state transition probability $P(S)$ and the probability given by the language model $P(W)$, namely $\boldsymbol{\eta}^{\text{lg}}\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) = \log P(S) + \log P(W)$, decoding with log-linear models becomes HMM Viterbi decoding (using combined state output score (3.66)), and it is equivalent to standard joint decoding discussed in section 2.2.

3.5.2 Segment Level Features

Various forms of frame level features were discussed in the previous subsection, where the features for utterances are comprised of feature based on frames. This is just one option for feature extraction. Rather than extracting features for each frame $\boldsymbol{o}_t$, feature generation can be based on segments $\boldsymbol{O}_{(i)}$. There are a number of advantages for this type of (segment level) features, e.g. the features could be expressed in a more compact form, and long-span dependencies within the segment are enabled [48]. In this subsection, the segment level features based on log-likelihoods will be discussed.

For segment level features, the general form of the features for an segment can be expressed as follows [65]:

$$\boldsymbol{\phi}\left(\boldsymbol{O}_{(i)}, w_i, \rho_i\right) = \begin{bmatrix} \delta(w_i, v_1)\varphi\left(\boldsymbol{O}_{(i)}\right) \\ \vdots \\ \delta(w_i, v_L)\varphi\left(\boldsymbol{O}_{(i)}\right) \end{bmatrix} \tag{3.67}$$

where $\{v_1, \ldots, v_L\}$ denote all possible sub-sentence units (such as words or tri-phones) in the dictionary, and $\delta(\cdot)$ is the Kronecker delta. $\varphi(\boldsymbol{O}_{(i)})$ is the feature vector for segment $\boldsymbol{O}_{(i)}$, and the form of the feature vector $\varphi(\boldsymbol{O}_{(i)})$ will be discussed in the following subsection. By substituting this form of segment level features (3.67) in, the acoustic features

[1] As described in equation (2.10), in standard joint decoding the combination weights are system dependent and do not vary with state j.

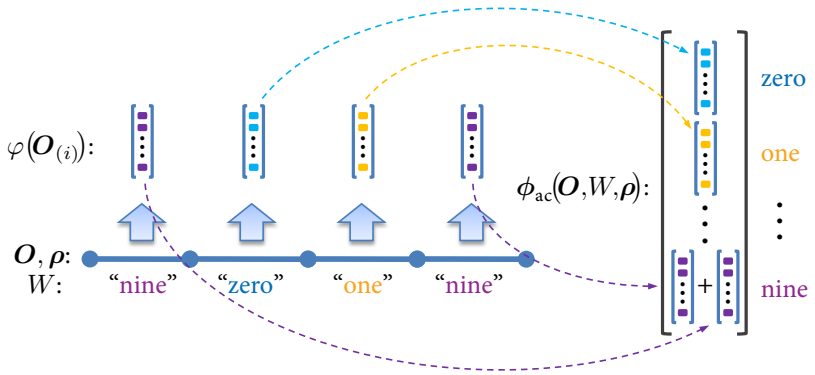


Figure 3.10: *Constructing the segment level acoustic features.*

defined in (3.56) can be further described as follows:

$$\phi_{ac}(\boldsymbol{O}, W, \boldsymbol{\rho}) = \sum_{i=1}^{I} \phi(\boldsymbol{O}_{(i)}, w_i, \rho_i) = \sum_{i=1}^{I} \begin{bmatrix} \delta(w_i, v_1)\varphi(\boldsymbol{O}_{(i)}) \\ \vdots \\ \delta(w_i, v_L)\varphi(\boldsymbol{O}_{(i)}) \end{bmatrix} \tag{3.68}$$

An example of constructing the segment level acoustic features is illustrated in Figure 3.10.

3.5.2.1 *Log-likelihood Features*

In speech recognition log-likelihood features are one of the most commonly used forms, and for this type of features $\varphi(\boldsymbol{O}_{(i)})$ can be expressed as follows [196, 209]:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p(\boldsymbol{O}_{(i)}|v_1) \\ \vdots \\ \log p(\boldsymbol{O}_{(i)}|v_L) \end{bmatrix} \tag{3.69}$$

where $p(\boldsymbol{O}_{(i)}|v_l)$ is the likelihood given by generative model associated with label v_l. When each segment corresponds a tri-phone[1], the number of all possible unique tri-phones L is very large. Then, the dimension of the feature vector $\varphi(\boldsymbol{O}_{(i)})$ is very high. This leads to inefficiency in training. Normally, when using tri-phones, the features $\varphi(\boldsymbol{O}_{(i)})$ only use log-likelihoods corresponding to the tri-phones having the same context with w_i (which is the label corresponds to $\boldsymbol{O}_{(i)}$). For example, when $w_i =$ "b-uh+k", only the models associated with the tri-phones having the form "b-$*$+k" (where $*$ denotes all possible mono-phones)

[1] The following discussion also applies to the systems using graphemes with context information.

are used in generating the log-likelihoods:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p(\boldsymbol{O}_{(i)}|\text{“b-aa+k”}) \\ \log p(\boldsymbol{O}_{(i)}|\text{“b-ae+k”}) \\ \vdots \\ \log p(\boldsymbol{O}_{(i)}|\text{“b-zh+k”}) \end{bmatrix} \tag{3.70}$$

This can significantly reduce the dimensionality of the features, and this form of setting can also be applied to the features based on multiple systems which will be discussed in the next subsection.

Let the parameters corresponding to features (3.68) be $\left[\boldsymbol{\eta}^{(v_1)\mathsf{T}}, \ldots, \boldsymbol{\eta}^{(v_L)\mathsf{T}}\right]^{\mathsf{T}}$. Analogous to the discussion for the frame level log-likelihood features, let the score $\boldsymbol{\eta}^{\text{lg}}\phi_{\text{lg}}(W, \boldsymbol{\rho})$ be the probability $P(W)$ given by the language model, the HMM probability can be retrieved by setting the elements of the parameters associated with the correct label to be 1 and others 0, namely $\boldsymbol{\eta}^{(v_1)} = \left[1, 0, \ldots, 0\right], \cdots, \boldsymbol{\eta}^{(v_L)} = \left[0, 0, \ldots, 1\right]$. This means the HMM baseline can be achieved by configuring the parameters in this fashion.

In addition to log-likelihoods, the derivatives of the log-likelihoods also can be used in the features:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p(\boldsymbol{O}_{(i)}|v_1) \\ \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}_{(i)}|v_1) \\ \vdots \\ \log p(\boldsymbol{O}_{(i)}|v_L) \\ \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}_{(i)}|v_L) \end{bmatrix} \tag{3.71}$$

where $\boldsymbol{\lambda}$ are the parameters of the generative model, and $\nabla_{\boldsymbol{\lambda}}$ represents the derivative with respect to $\boldsymbol{\lambda}$. There are a number of advantage to using derivative features, e.g. if the generative model is an HMM then the resulting features do not have the same underlying conditional independence assumptions of the HMM [65, 209], moreover the derivative features can provide more discriminative information [109, 148, 209].

3.5.2.2 *Features Based on Multiple Systems*

In section 3.5.1.2 the frame level features based on multiple systems were discussed. Analogously, the segment level features also can be based on multiple systems. Let D be the number of systems used in feature generation. One form of the segment level features $\varphi(\boldsymbol{O}_{(i)})$

can be expressed as follows [38]:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p_1(\boldsymbol{O}_{(i)}|v_1) \\ \vdots \\ \log p_1(\boldsymbol{O}_{(i)}|v_L) \\ \vdots \\ \log p_D(\boldsymbol{O}_{(i)}|v_1) \\ \vdots \\ \log p_D(\boldsymbol{O}_{(i)}|v_L) \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} \text{features from system 1} \\ \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} \text{features from system } D \end{matrix} \tag{3.72}$$

This type of features has been studied in work [38], where two systems were used in feature generation. Since the features use the log-likelihoods not only from the correct labels, but also from the competing ones, the dimensionality of the feature vector is very high, and this leads to inefficiency in training. Alternatively, one simplified form can be used [199]:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p_1(\boldsymbol{O}_{(i)}|w_i) \\ \vdots \\ \log p_D(\boldsymbol{O}_{(i)}|w_i) \end{bmatrix} \tag{3.73}$$

where only the log-likelihoods associated with the correct labels w_i are used in feature generation. In this thesis, this type of features (3.73) will be examined in the experimental section. The commonly used acoustic features are tabulated in Table 3.1.

3.5.3 *Language Features*

The form of the language features are mainly associated with the state, phone and word sequence [65]. When using frame level features, the segmentation also specifies the state information. Then one of the simplest forms for language features is based on the state transition features [79]:

$$\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) = \sum_{t=1}^{T} \begin{bmatrix} \vdots \\ \delta(s_t = i) \\ \delta(s_t = i, s_{t-1} = j) \\ \vdots \end{bmatrix}, \qquad \forall i, j \tag{3.74}$$

where T is the number of frames. i and j denote any possible states pair which specifies a valid state transition.

Feature Type		Representation	Papers
Frame Level	Gaussian Statistics	$\varphi(\boldsymbol{o}_t) = \begin{bmatrix} 1 \\ \boldsymbol{o}_t \\ \text{diag}(\boldsymbol{o}_t \boldsymbol{o}_t^{\mathsf{T}}) \end{bmatrix}$	[79]
	Derivatives	$\varphi(\boldsymbol{o}_t) = \begin{bmatrix} \vdots \\ \log p(\boldsymbol{o}_t \vert s) \\ \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{o}_t \vert s) \\ \vdots \end{bmatrix}, \forall s$	[109, 148]
	from Multi-Systems	$\varphi(\boldsymbol{o}_t) = \begin{bmatrix} \log p_1(\boldsymbol{o}_t \vert s_t) \\ \vdots \\ \log p_D(\boldsymbol{o}_t \vert s_t) \end{bmatrix}$	[199]
Segment Level	Derivatives	$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \vdots \\ \log p(\boldsymbol{O}_{(i)} \vert v) \\ \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}_{(i)} \vert v) \\ \vdots \end{bmatrix}, \forall v$	[148]
	from Multi-Systems	$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p_1(\boldsymbol{O}_{(i)} \vert w_i) \\ \vdots \\ \log p_D(\boldsymbol{O}_{(i)} \vert w_i) \end{bmatrix}$	[198, 199]

Table 3.1: *The commonly used acoustic features.*

For segment level features, the corresponding language features can be based on unigram and bigram features:

$$\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) = \sum_{i=1}^{I} \begin{bmatrix} \vdots \\ \delta(w_i = v) \\ \delta(w_i = v, w_{i-1} = v') \\ \vdots \end{bmatrix}, \qquad \forall v, v' \tag{3.75}$$

where I is the number of segments. v and v' denotes any possible sub-sentence units, such as words, in the dictionary. Another extensively used form of the language features is based on the probability given by the language model [209]:

$$\boldsymbol{\phi}_{\text{lg}}(W, \boldsymbol{\rho}) = \begin{bmatrix} \log P(W) \end{bmatrix} \tag{3.76}$$

where $P(W)$ is the probability of the word sequence W given by the language model, e.g. the n-gram language model. This type of language features (3.76) will be examined in the experimental section.

3.6 Summary

In this chapter, various unstructured discriminative models have been introduced. For unstructured models, the structure of the class labels is not considered and each class label is treated as a single (atomic) unit, these models cannot be directly applied to continuous speech recognition. Thus, the framework of acoustic code breaking was discussed, where the continuous speech is segmented into segments, and then each segment is treated independently and classified separately. Rather than treating different speech segments independently, structured discriminative models can be employed in continuous speech recognition directly. In structured models the label structure is considered, e.g. different sentences share the same common set of sub-sentence units (such as words or phones). Since the features play a significant role in discriminative models, different forms of features were also discussed. In general these features can be divided up into the acoustic features and the language features, and various forms of the acoustic features are summarised in Table 3.1.

Chapter 4

Bayesian Non-parametric Models

In the previous sections, various commonly used parametric models (including both the generative and discriminative models) in speech recognition were discussed, where $\boldsymbol{o}$ denotes a speech observation vector, $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ indicate the model parameters for generative and discriminative models in speech recognition. In this section, a general discussion on Bayesian non-parametric models will be given, and a more general form of notations will be used, e.g. $\boldsymbol{x}$ denotes the input vector, and $\mathcal{G}$ indicates the whole model parameter set.

4.1 Motivations

Maximum likelihood (ML) estimation [89] discussed in section 2.3.1 is one of the most commonly used estimate methods for parametric models. In ML estimation, the model parameters are considered as fixed but unknown values, and the parameters can be estimated by maximising the likelihood function. For example, given a parametric model $p(\boldsymbol{x}|\mathcal{G})$ with training data $\mathcal{D} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, the model parameters $\mathcal{G}$ can be estimated by maximising the likelihood function:

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}} \prod_n p(\boldsymbol{x}_n|\mathcal{G}) \tag{4.1}$$

In maximum likelihood estimation, since the model parameters are estimated by maximising the likelihood given the training data, extreme conclusions might be drawn, espe-

cially when the number of training data is relatively small. This is also known as over-fitting. In order to mitigate this problem, Bayesian approaches can be employed. In Bayesian approaches, the model parameters are considered as random variables, and a prior distribution $p(\mathcal{G})$ is used to express the uncertainty of these parameters. When making predictions, all the model parameters are marginalised out:

$$p(\boldsymbol{x}|\mathcal{D}) = \int p(\boldsymbol{x}|\mathcal{G})p(\mathcal{G}|\mathcal{D})d\mathcal{G} \tag{4.2}$$

This is called the *predictive distribution*, which is the distribution of the unobserved data (prediction) conditional on the observed ones [166]. In equation (4.2), the posterior distribution $p(\mathcal{G}|\mathcal{D})$ can be obtained through Bayes' rule:

$$p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G}) \prod_n p(\boldsymbol{x}_n|\mathcal{G}) \tag{4.3}$$

When more data are observed, the posterior distribution can be viewed as the prior, and the new posterior distribution can be obtained through Bayes' rule described in (4.3). This motivates *maximum a posteriori* (MAP) to be an adaptation approach in speech recognition [66, 68]. In MAP estimation, the model parameters can be estimated by maximising the posterior distribution described in (4.3).

In parametric approaches, the complexity of the model needs to be determined in advance, e.g. specifying the number of states in a hidden Markov model. In order to avoid the problem of setting model complexity, Bayesian non-parametric models can be applied, e.g. the infinite Gaussian mixture model (iGMM) was proposed to sidestep the problem of choosing component number in GMMs [149]. For parametric models, the number of parameters is finite and predetermined. In contrast, for non-parametric models, the number of parameters is infinite, namely, the size of the parameter set $\mathcal{G}$ may be infinite. In Bayesian non-parametric models, a prior distribution is defined on the infinite dimensional parameter space. Given the training data (or observed data), data analysis is performed by *posterior inference*, computing the posterior distribution of the model parameters given the observed data. Rather than specifying the model complexity in advance (parametric models), the model complexity is part of the posterior inference for Bayesian non-parametric models. When making predictions, the posterior distribution of the model parameters can be integrated over, effectively averaging over models of all possible complexity [50, 70, 133].

Suppose the model complexity is denoted by M, e.g. the number of components in the mixture model, the predictive distribution for Bayesian non-parametric models can be

obtained by marginalising over all the model parameters (including the model complexity). This might be expressed as follows:

$$p(\boldsymbol{x}|\mathcal{D}) = \sum_{M} \int p(\boldsymbol{x}|\mathcal{G}_M, M)p(\mathcal{G}_M, M|\mathcal{D})d\mathcal{G}_M \tag{4.4}$$

where $\mathcal{G}_M$ denotes the model parameters associated with model complexity M. Since the complexity of the model is considered, the posterior distribution of the model parameters $p(\mathcal{G}_M, M|\mathcal{D})$ is extremely complicated or does not have an analytical form. This leads to intractability in calculating the predictive distribution (4.4). Thus, approximate methods need to be applied, e.g. the *Monte Carlo* (MC) approaches [6, 16].

4.1.1 De Finetti's Theorem

In the previous section, the motivation for Bayesian non-parametric models was discussed. This section will examine de Finetti's theorem [45], which is often taken as a justification for Bayesian non-parametric models.

Consider a sequence of variables $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$. These variables are *exchangeable*, if any permutation of their indices has equal probability:

$$p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N) = p(\boldsymbol{x}_{\sigma(1)}, \cdots, \boldsymbol{x}_{\sigma(N)}) \tag{4.5}$$

where $\{\sigma(1), \cdots, \sigma(N)\}$ is any permutation (or reordering) of $\{1, \cdots, N\}$. This definition (4.5) can be extended to the infinite situation. An infinite sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots\}$ is *infinite exchangeable*, if any finite subset of the variables is exchangeable [11, 69]. Exchangeability is a reasonable assumption, since it is common that the indices of variables are only chosen to distinguish from each other. Exchangeability reflects the assumption that the variables do not depend on their indices although they might be dependent on each other. Moreover, infinite exchangeability makes the model unaffected by the unobserved data (e.g. the test data). Compared with the independently and identically distributed (i.i.d.) assumption, exchangeability is a much weaker assumption, and the i.i.d. assumption automatically results in exchangeability of the sequence [133, 168, 177].

De Finetti's theorem states that for any infinity exchangeable sequence of variables $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots\}$, where $\boldsymbol{x}_n \in \mathcal{X}$, there exists a random variable set $\mathcal{G}$, such that the joint distribu-

tion of any N $(N \geq 1)$ variables satisfies:

$$p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N) = \int p(\mathcal{G}) \prod_{n=1}^{N} p(\boldsymbol{x}_n|\mathcal{G}) d\mathcal{G} \tag{4.6}$$

where $p(\mathcal{G})$ is the prior distribution over parameters $\mathcal{G}$. In de Finetti's theorem, when $\mathcal{X}$ is a finite dimensional discrete space, $\mathcal{G}$ are finite-dimensional. When $\mathcal{X}$ is a continuous space, $\mathcal{G}$ are infinite-dimensional (typically a random measure).

According to de Finetti's theorem described in (4.6), exchangeability automatically implies the existence of a Bayesian model with random latent parameters $\mathcal{G}$. Thus, in Bayesian models the assumption, that is the existence of randomly distributed parameters, is not a modelling hypothesis, but a mathematical consequence of the data's properties [133]. When the observations are continuous, the parameters $\mathcal{G}$ are infinite-dimensional, since there is no finite parameterisation for the space of continuous densities [133, 168]. This implies the number of parameters in the Bayesian model is infinite. The Bayesian model then becomes a Bayesian non-parametric model. Thus, de Finetti's theorem is considered as the justification of Bayesian non-parametric models.

In de Finetti's theorem, the model parameter set $\mathcal{G}$ is typically a random measure. The *Dirichlet process* (DP) [18, 44, 160] defines a distribution over probability measures with many attractive properties, and is widely used in practice due to its simplicity and the computational efficiency in inference. This motivates a class of Bayesian non-parametric models based on Dirichlet processes which will be further discussed in the following sections.

4.2 Bayesian Approaches

In the previous section, the motivation for employing Bayesian non-parametric models and de Finetti's theorem were discussed. This section will briefly introduce Bayesian approaches for generative models and discriminative models.

4.2.1 Bayesian Inference

In terms of a generative model, a frequentist point of view may be adopted, where the model parameters are considered as fixed but unknown, and the values of the parameters are determined by a estimator (or criterion), e.g. the maximum likelihood (ML) estimator described

in (4.1). These estimated parameters then can be used in making predictions. As discussed in the previous section, ML estimation might lead to over-fitting. Bayesian approaches are an alternative. Compared with frequentist schemes, Bayesian approaches employ a prior distribution over the model parameters to express the uncertainty of these parameters before observing the data, and make precise revisions of uncertainty in the light of new evidence. The revised distribution is expressed in the form of a posterior distribution [16, 81]. As described in (4.2), when making predictions, all the model parameters are marginalised out:

$$p(\boldsymbol{x}|\mathcal{D}) = \int p(\boldsymbol{x}|\mathcal{G})p(\mathcal{G}|\mathcal{D})d\mathcal{G} \tag{4.7}$$

This is called the *predictive distribution* [166]. For a generative model, the posterior distribution $p(\mathcal{G}|\mathcal{D})$ in equation (4.7) can be obtained according to Bayes' rule:

$$p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G}|\boldsymbol{\Lambda}) \prod_n p(\boldsymbol{x}_n|\mathcal{G}) \tag{4.8}$$

where $p(\mathcal{G}|\boldsymbol{\Lambda})$ is a prior distribution with hyperparameters $\boldsymbol{\Lambda}$. It is worth noting that the maximum likelihood estimate is the maximum of the posterior distribution described in (4.8) without considering the prior distribution $p(\mathcal{G}|\boldsymbol{\Lambda})$ (or when the prior is a uniform distribution). In ML estimation, the likelihood function is employed to estimate the model parameters, whereas Bayesian approaches utilise the likelihood function to update the prior beliefs.

In the posterior distribution described in (4.8), the prior distribution $p(\mathcal{G}|\boldsymbol{\Lambda})$ captures any available knowledge about the data generation process. The hyperparameters $\boldsymbol{\Lambda}$ can be set to some fixed value based on our prior beliefs [168]. A *fully Bayesian analysis* places a prior distribution $p(\boldsymbol{\Lambda})$ on the hyperparameters. In practice, an *empirical Bayesian method* is often applied, in which the hyperparameters $\boldsymbol{\Lambda}$ are estimated by maximising the marginal likelihood of the training data:

$$\hat{\boldsymbol{\Lambda}} = \arg\max_{\boldsymbol{\Lambda}} \int p(\mathcal{G}|\boldsymbol{\Lambda}) \prod_n p(\boldsymbol{x}_n|\mathcal{G})d\mathcal{G} \tag{4.9}$$

As described in equation (4.7), the predictive distribution is obtained by integrating over all the model parameters. The posterior distribution described in the right hand side of (4.8) is un-normalised. When the normalisation term of the posterior distribution $p(\mathcal{G}|\mathcal{D})$ is not possible to compute, or the posterior distribution does not have a closed form, the

integral by definition in calculating the predictive distribution (4.7) becomes intractable. Then, approximate methods such as Monte Carlo (MC) approaches [6] can be applied. In the MC approach, the integral in the predictive distribution (4.7) can be approximated by summing over K samples:

$$p(\boldsymbol{x}|\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{G}^{(k)}) \tag{4.10}$$

where the samples $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(K)}\}$ are drawn from the posterior distribution $p(\mathcal{G}|\mathcal{D})$ described in (4.8). Alternative to MC approaches [6], a point estimate scheme can be applied, in which the model parameters $\mathcal{G}$ are estimated by maximising the model posterior distribution described in (4.8):

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}} p(\mathcal{G}|\mathcal{D}) = \arg\max_{\mathcal{G}} p(\mathcal{G}|\boldsymbol{\Lambda}) \prod_{n} p(\boldsymbol{x}_n|\mathcal{G}) \tag{4.11}$$

This is called *maximum a posteriori* (MAP) estimation. Compared with ML estimation described in (4.1), a prior distribution is incorporated in MAP estimation. Thus, MAP estimation can be viewed as a regularisation of ML estimation. Given the point estimated parameters $\hat{\mathcal{G}}$, the posterior distribution of the model parameters can be written in the form of a Dirac delta function, namely $p(\mathcal{G}|\mathcal{D}) \approx \delta(\mathcal{G}, \hat{\mathcal{G}})$. Then, the predictive distribution described in (4.7) becomes:

$$p(\boldsymbol{x}|\mathcal{D}) \approx \int p(\boldsymbol{x}|\mathcal{G})\delta(\mathcal{G}, \hat{\mathcal{G}})d\mathcal{G} = p(\boldsymbol{x}|\hat{\mathcal{G}}) \tag{4.12}$$

4.2.2 *Conditional Bayesian Inference*

For a classification task, a discriminative model, which models the posterior distribution $P(w|\boldsymbol{x})$ directly, might be preferred. One reason for selecting discriminative models is that they do not make any assumption on the distribution of the input data, and it is not necessary to model the density $p(\boldsymbol{x}|w)$, Instead these models focus on the boundary between classes. A simple example is illustrated in Figure 3.1 of Chapter 3. The complicated structure in the probability density function has little effect on the posterior probabilities. Therefore, it is not always necessary to compute the joint distribution. This is the main reason why discriminative models have been widely and successfully used [105, 132, 183].

For the discriminative model $P(w|\boldsymbol{x}, \mathcal{G})$, which gives the conditional distribution of the class w given the observation $\boldsymbol{x}$, with model parameters $\mathcal{G}$, assume the training data are $\mathcal{D} = \{(\boldsymbol{x}_1, w_1), \ldots, (\boldsymbol{x}_N, w_N)\}$, ML estimation for the generative model described in equation (4.1) becomes conditional maximum likelihood (CML) estimation:

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}} \prod_n P(w_n|\boldsymbol{x}_n, \mathcal{G}) \tag{4.13}$$

Under a Bayesian setting, the model parameters $\mathcal{G}$ are given a prior distribution, and Bayes' rule is used to update the distribution after new evidence is observed. When making predictions, similar to the predictive distribution for a generative model described in (4.7), the *class posterior distribution*[1] for the discriminative model can be obtained by marginalising out all the model parameters [17, 105]:

$$P(w|\boldsymbol{x}, \mathcal{D}) = \int P(w|\boldsymbol{x}, \mathcal{G}) p(\mathcal{G}|\mathcal{D}) d\mathcal{G} \tag{4.14}$$

Analogous to the posterior distribution of model parameters for a generative model described in (4.8), the posterior distribution $p(\mathcal{G}|\mathcal{D})$ can be obtained according to Bayes' rule:

$$p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G}|\boldsymbol{\Lambda}) \prod_n P(w_n|\boldsymbol{x}_n, \mathcal{G}) \tag{4.15}$$

where $p(\mathcal{G}|\boldsymbol{\Lambda})$ is the prior distribution of the model parameters, and $\boldsymbol{\Lambda}$ are hyperparameters. Normally, neither the posterior distribution $p(\mathcal{G}|\mathcal{D})$ nor the integral in the class posterior distribution (4.14) is tractable. Analogous to the approximation made for generative models (4.10), Monte Carlo (MC) approaches can be applied to approximate the class posterior distribution (4.14):

$$P(w|\boldsymbol{x}, \mathcal{D}) \approx \frac{1}{K} \sum_{k=1}^{K} P(w|\boldsymbol{x}, \mathcal{G}^{(k)}) \tag{4.16}$$

where the samples $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(K)}\}$ are drawn from the posterior distribution $p(\mathcal{G}|\mathcal{D})$ described in (4.15).

As discussed in section 4.2.1, in addition to MC approaches, MAP estimation (4.11) (a point estimate scheme) can also be applied. MAP estimation for discriminative models can be described as:

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}} p(\mathcal{G}|\mathcal{D}) = \arg\max_{\mathcal{G}} p(\mathcal{G}|\boldsymbol{\Lambda}) \prod_n p(w_n|\boldsymbol{x}_n, \mathcal{G}) \tag{4.17}$$

[1] To distinguish from the predictive distribution of the generative model, the predictive distribution of the discriminative model is called the class posterior distribution throughout this thesis.

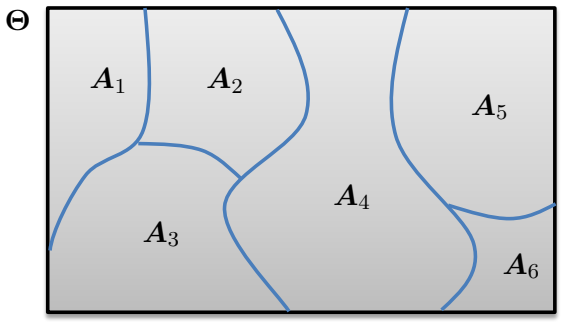


Figure 4.1: *A possible partition* $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_6\}$ *of the set* $\boldsymbol{\Theta}$.

Given the point estimated parameters $\hat{H}$, the posterior distribution of the model parameters can be written in the form of a Dirac delta function, namely $p(\mathcal{G}|\mathcal{D}) \approx \delta(\mathcal{G}, \hat{\mathcal{G}})$. Then, the class posterior distribution described in (4.14) can be further written as:

$$P(w|\boldsymbol{x}, \mathcal{D}) \approx \int P(w|\boldsymbol{x}, \mathcal{G})\delta(\mathcal{G}, \hat{\mathcal{G}})d\mathcal{G} = P(w|\boldsymbol{x}, \hat{\mathcal{G}}) \tag{4.18}$$

4.3 Dirichlet Processes

In the previous sections, the basic ideas of Bayesian inference were introduced. This section will discuss the *Dirichlet process* (DP) [44], which is a distribution over distributions with a wide support that is the space of all (discrete) distributions. The Dirichlet process is one of the most widely used stochastic processes in Bayesian non-parametric models due to its simplicity, wide coverage of the distributions, and tractability in posterior inference. However, the formal definition of the Dirichlet process does not provide a mechanism to sample from this process. Two practical representations of the Dirichlet process, called the *stick-breaking construction* [160] and the *Chinese restaurant process* (CRP) [138], will be introduced in this section. The stick-breaking process and Chinese restaurant process provide mechanisms to draw samples and predict future observations from the Dirichlet process. Since the Dirichlet process is discrete, which is too limited to model the continuous data, the infinite mixture models based on the two practical representations of the Dirichlet process will also be discussed.

4.3.1 The Definition of the Dirichlet Process

Let $\boldsymbol{\Theta}$ be the set of all possible outcomes (which is sometimes called the *sample space*), e.g. heads and tails when tossing a coin, a finite measurable partition $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_L\}$ on $\boldsymbol{\Theta}$ can be defined as:

$$\bigcup_{l=1}^{L} \boldsymbol{A}_l = \boldsymbol{\Theta} \qquad \boldsymbol{A}_l \cap \boldsymbol{A}_i = \varnothing \qquad \forall i \neq l \tag{4.19}$$

where $\varnothing$ is the empty set. One possible partition of the set $\boldsymbol{\Theta}$ is illustrated in Figure 4.1. Some complimentary knowledge on probability measures is given in Appendix A.

Let $\boldsymbol{G}_0$ be a probability distribution[1] on $\boldsymbol{\Theta}$, $\boldsymbol{G}$ be a random measure of this set $\boldsymbol{\Theta}$, and α be a positive real number. If the measure $\boldsymbol{G}$ on any finite measurable partition $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_L\}$ is Dirichlet distributed:

$$\boldsymbol{G}(\boldsymbol{A}_1), \cdots, \boldsymbol{G}(\boldsymbol{A}_L) \sim \text{Dirichlet}(\alpha \boldsymbol{G}_0(\boldsymbol{A}_1), \cdots, \alpha \boldsymbol{G}_0(\boldsymbol{A}_L)) \text{ [2]} \tag{4.20}$$

Then the random measure $\boldsymbol{G}$ is drawn from a Dirichlet process with *concentration parameter* α and *base distribution* $\boldsymbol{G}_0$ [44]:

$$\boldsymbol{G} \sim \text{DP}(\alpha, \boldsymbol{G}_0) \tag{4.21}$$

The Dirichlet process defines a distribution over probability measures. For any measurable set $\boldsymbol{A} \in \boldsymbol{\Theta}$, the expectation over the Dirichlet process satisfies $E(\boldsymbol{G}(\boldsymbol{A})) = \boldsymbol{G}_0(\boldsymbol{A})$. Thus, the base distribution $\boldsymbol{G}_0$ gives the mean of a Dirichlet process, while the concentration parameter α can be considered as the precision (inverse of variance), which determines how the sampled distribution $\boldsymbol{G}$ deviates from the base distribution $\boldsymbol{G}_0$ on average [177].

4.3.1.1 The Posterior Distribution

In the previous section the formal definition of the Dirichlet process was introduced. The posterior distribution of the Dirichlet process will be discussed in this section. And this posterior distribution motivates two practical representations of the Dirichlet process (called

[1] In measure theory, a probability distribution is a probability measure, and this distribution (or probability measure) can be specified by a probability function.

[2] If the probability distribution $\boldsymbol{G}_0$ is specified by a probability density function $p(\boldsymbol{\theta})$ (continuous), $\boldsymbol{G}_0(\boldsymbol{A}_l)$ can be described as $\boldsymbol{G}_0(\boldsymbol{A}_l) = \int_{\boldsymbol{\theta} \in \boldsymbol{A}_l} p(\boldsymbol{\theta}) d\boldsymbol{\theta}$; If $\boldsymbol{G}_0$ is specified by a probability mass function $\sum_i \pi_i \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$ (discrete), $\boldsymbol{G}_0(\boldsymbol{A}_l)$ can be described as $\boldsymbol{G}_0(\boldsymbol{A}_l) = \sum_{\boldsymbol{\theta}_i \in \boldsymbol{A}_l} \pi_i$.

the stick-breaking process and the Chinese restaurant process), which will be discussed in the following subsections.

Assume $\boldsymbol{G} \sim \mathrm{DP}(\alpha, \boldsymbol{G}_0)$ is sampled from the Dirichlet process. Since $\boldsymbol{G}$ itself is a distribution, samples can be drawn from $\boldsymbol{G}$. Let $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$[1] be a sequence of samples from the random measure $\boldsymbol{G}$, and $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_L\}$ be a finite partition of the set $\boldsymbol{\Theta}$. Given the Dirichlet distribution induced by a finite partition described in (4.20) and the conjugacy of the Dirichlet distribution, the posterior distribution is also Dirichlet distributed [168, 177]:

$$\big(\boldsymbol{G}(\boldsymbol{A}_1), \cdots, \boldsymbol{G}(\boldsymbol{A}_L)\big) | \boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N \sim \mathrm{Dirichlet}\big(\alpha \boldsymbol{G}_0(\boldsymbol{A}_1) + N_1, \cdots, \alpha \boldsymbol{G}_0(\boldsymbol{A}_L) + N_L\big) \tag{4.22}$$

where N_l is the number of samples in $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$ drawn from $\boldsymbol{A}_l$. Since (4.22) is true for any finite measurable partition, according to the definition of the Dirichlet process, the posterior distribution over $\boldsymbol{G}$ also follows a Dirichlet process [168, 177]:

$$\boldsymbol{G} | \boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N \sim \mathrm{DP}\left(N + \alpha, \frac{1}{N + \alpha}\Big(\sum_{n=1}^{N} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}'_n) + \alpha \boldsymbol{G}_0\Big)\right) \tag{4.23}$$

where $\delta(\boldsymbol{\theta}, \boldsymbol{\theta}'_n)$ is a point mass at $\boldsymbol{\theta}'_n$, which is a Dirac delta function. According to the posterior distribution of $\boldsymbol{G}$ in (4.23), as the number of observations grow, when $N \gg \alpha$, the posterior distribution is dominated by the empirical distribution $\frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}'_n)$, which is an approximation of the true underlying distribution. This means the posterior distribution of the random measure approaches the true underlying distribution [177].

4.3.2 Stick-breaking Processes

In the previous subsection, the definition of the Dirichlet process was introduced. However, this definition does not provide a mechanism to draw samples from the Dirichlet process. In this section, a representation of the Dirichlet process called the stick-breaking process [160], that allows samples to be drawn, will be discussed in detail.

[1] Since $\boldsymbol{G}$ is discrete (which will be discussed in the following subsection), different $\boldsymbol{\theta}'_n$ may have identical value; The unique values of $\{\boldsymbol{\theta}'_1, \ldots, \boldsymbol{\theta}'_N\}$ are denoted as $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$.

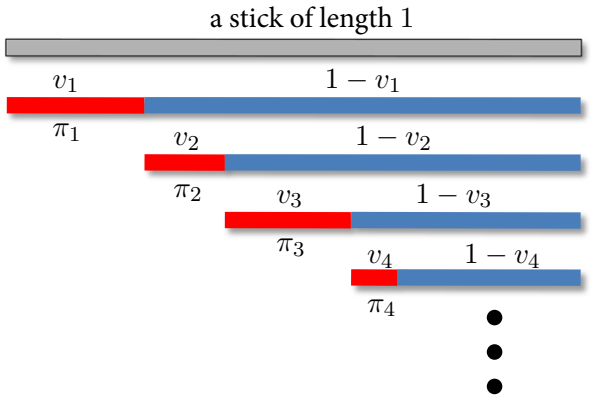


Figure 4.2: *The stick-breaking process. The red parts are to be broken off, and the lengths of the red parts of the stick correspond to the weights from a Dirichlet process.*

Given the posterior distribution of the measure (4.23), which is a Dirichlet process, then the expectation of the measure $\boldsymbol{G}$ is:

$$E\left(\boldsymbol{G}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\right) = \frac{1}{N+\alpha}\left(\sum_{n=1}^{N} \delta\left(\boldsymbol{\theta}, \boldsymbol{\theta}'_n\right) + \alpha \boldsymbol{G}_0\right) \tag{4.24}$$

when the number of observations goes to infinity, the second term in brackets becomes 0, then the following can be derived:

$$\lim_{N\to\infty} E\left(\boldsymbol{G}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\right) = \sum_{m=1}^{\infty} \pi_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) \tag{4.25}$$

where $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots\}$ are the unique values in $\{\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2, \cdots\}$, and π_m is the limiting frequency of $\boldsymbol{\theta}_m$ (when $N \to \infty$). According to the expected measure (4.25), sampled measures from the Dirichlet process are discrete with probability one. This is verified by the stick-breaking construction of the Dirichlet process discussed in the rest of this subsection.

The stick-breaking process provides a constructive representation of the Dirichlet process. In the stick breaking process, a sequence of weights are sampled:

$$\begin{aligned} v_m &\sim \text{Beta}(1, \alpha) \\ \pi_m &= v_m \prod_{i=1}^{m-1} (1 - v_i) \end{aligned} \tag{4.26}$$

where α is the parameter of the *Beta distribution* [16], which arises from the marginal distribution of the Dirichlet process. The stick-breaking process is illustrated in Figure 4.2.

Consider a stick with length 1. π_m is the length of the part to be broken from the stick in the mth breaking, and v_m denotes the ratio of the remaining stick to be broken. By recursively breaking this stick, a sequence of weights $\{\pi_1, \pi_2, \cdots\}$ can be obtained. Normally, the weights $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots\}$ from the stick-breaking process are denoted as $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$, which is named after Griffiths, Egnen and McCloskey [138].

Given the base distribution $\boldsymbol{G}_0$, the stick-breaking construction of the Dirichlet process can be described as follows[1]:

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{GEM}(\alpha) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{G}_0 \\ \boldsymbol{G} &= \sum_{m=1}^{\infty} \pi_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) \end{aligned} \tag{4.27}$$

where $\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)$ is the Dirac delta function. The stick-breaking construction guarantees $\boldsymbol{G} \sim \text{DP}(\alpha, \boldsymbol{G}_0)$. According to the stick-breaking construction of the Dirichlet process, samples from the Dirichlet process are discrete measures with probability one [160, 168].

4.3.3 Chinese Restaurant Processes

In the previous subsection, the stick breaking process was introduced, which provides a constructive representation of the Dirichlet process. In this subsection, another practical presentation called the *Chinese restaurant process* (CRP) will be discussed.

The Chinese restaurant process is a metaphor which assumes there are infinite number of tables in a restaurant. When a customer come to the restaurant, the probability of sitting at an occupied table is proportional to the number of people already sitting there, and the probability of sitting at a new table is proportional to α:

$$P\big(z_{N+1} = m | z_1, \cdots, z_N, \alpha\big) = \begin{cases} \dfrac{N_m}{N+\alpha} & \text{where } m \text{ is an occupied table} \\ \dfrac{\alpha}{N+\alpha} & \text{where } m \text{ is an unoccupied table} \end{cases} \tag{4.28}$$

where $N+1$ is the $(N+1)$th costumer, m is the mth table, indicator z_{N+1} denotes the $(N+1)$th costumer sitting at the z_{N+1}th table, and N_m is the number of people (except

[1] Here, $\boldsymbol{G}_0$ is considered as a continuous distribution, to distinguish the samples $\boldsymbol{\theta}'_i$ drawn from $\boldsymbol{G}$ (which is discrete as described in (4.27)), the sample drawn from $\boldsymbol{G}_0$ is denoted as $\boldsymbol{\theta}_m$.

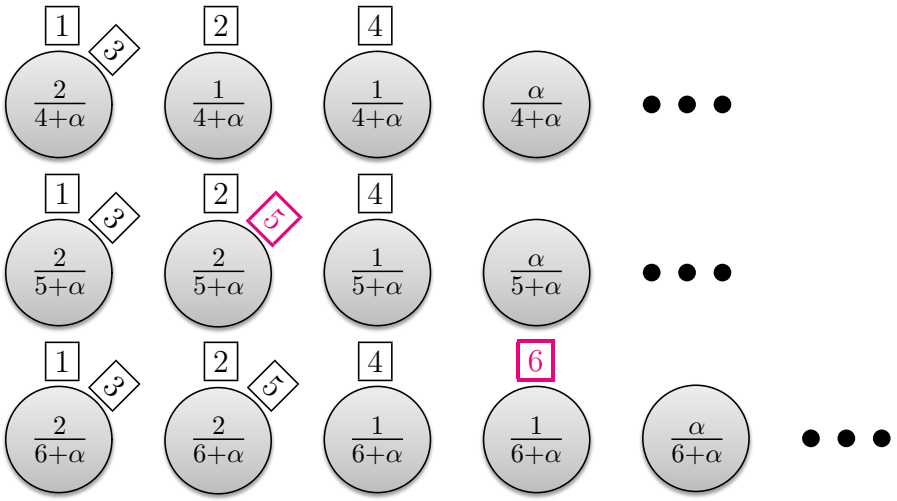


Figure 4.3: *The Chinese restaurant process. In this figure, each row gives a status of the restaurant. When a new customer comes to this restaurant, the probability of sitting at an occupied table is proportional to the number of people already sitting there, and the probability of sitting at a new table is proportional to α. These probabilities are given in each circle of this figure.*

the $(N+1)$th person) occupying the mth table. Normally, the indicators $\boldsymbol{z} = \{z_1, z_2, \cdots\}$ sampled from the Chinese restaurant process are denoted as $\boldsymbol{z} \sim \text{CRP}(\alpha)$. An example of the Chinese restaurant process is illustrated in Figure 4.3. The derivation of this process will be detailed in the rest of this subsection.

Let $\boldsymbol{G}$ be a random measure drawn from $\text{DP}(\alpha, \boldsymbol{G}_0)$, and $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$ a sequence of samples drawn from $\boldsymbol{G}$. Given the posterior distribution of $\boldsymbol{G}$, which is still a Dirichlet process as described in (4.23), the predictive distribution of $\boldsymbol{\theta}'_{N+1}$ conditional on $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$ can be obtained by marginalising out $\boldsymbol{G}$. Since $P(\boldsymbol{\theta}|\boldsymbol{G}, \boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N) = \boldsymbol{G}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N$, then the following can be derived:

$$\begin{aligned} P\big(\boldsymbol{\theta}'_{N+1} = \boldsymbol{\theta}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\big) &= \int P(\boldsymbol{\theta}|\boldsymbol{G}, \boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N) p(\boldsymbol{G}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N) d\boldsymbol{G} \\ &= E\big(\boldsymbol{G}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\big) \end{aligned} \tag{4.29}$$

Given the definition of the expected measure (4.24), equation (4.29) then can be further written as:

$$P\big(\boldsymbol{\theta}'_{N+1} = \boldsymbol{\theta}|\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\big) = \frac{1}{N+\alpha}\Big(\sum_{n=1}^{N} \delta\big(\boldsymbol{\theta}, \boldsymbol{\theta}'_n\big) + \alpha \boldsymbol{G}_0\Big) \tag{4.30}$$

The predictive distribution of $\boldsymbol{\theta}'_{N+1}$ is the base distribution of the posterior Dirichlet process (4.23), which is the posterior distribution of the random measure $\boldsymbol{G}$ given $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$.

The process of obtaining $\{\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2, \cdots\}$ according to the predictive distributions (4.30) is known as the Pólya urn scheme [18], which is a metaphor to help interpreting this process. Specifically, the Pólya urn scheme can be described as the follows. There is no ball in the urn at the beginning, a ball with colour $\boldsymbol{\theta}'_1$ drawn from $\boldsymbol{G}_0$ is put in the urn. In the following steps, take the $(N+1)$th step for example, with probability $\frac{\alpha}{N+\alpha}$, a new ball having colour $\boldsymbol{\theta}'_{N+1}$ drawn from $\boldsymbol{G}_0$ is put in the urn; With probability $\frac{N}{N+\alpha}$, a ball having colour $\boldsymbol{\theta}'_{N+1}$ is drawn from the urn, then the ball is replaced to the urn with an extra ball having the same colour $\boldsymbol{\theta}'_{N+1}$.

According to the stick-breaking construction of the Dirichlet process (4.27), the random measure $\boldsymbol{G}$ drawn from the Dirichlet process is discrete. Thus, the observations $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$ drawn from $\boldsymbol{G}$ have positive probability taking identical values. This leads to the clustering property of the Dirichlet process, as shown in the Pólya urn scheme. Let $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M\}$ be the unique values of $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$, and N_m denote the number of value $\boldsymbol{\theta}_m$ in $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$, which satisfies $\sum_{m=1}^{M} N_m = N$. Then the predictive distribution (4.30) can be rewritten as:

$$P(\boldsymbol{\theta}'_{N+1} = \boldsymbol{\theta} | \boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N) = \frac{1}{N+\alpha}\Big(\sum_{m=1}^{M} N_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) + \alpha \boldsymbol{G}_0\Big) \tag{4.31}$$

By introducing the indicator variable z_n, which denotes the cluster (or component) associated with the nth observation, the predictive distribution (4.31) can be expressed as the following form:

$$P(z_{N+1} = z | z_1, \cdots, z_N, \alpha) = \frac{1}{N+\alpha}\Big(\sum_{m=1}^{M} N_m \delta(z, m) + \alpha \delta(z, M+1)\Big) \tag{4.32}$$

where $M+1$ denotes a new cluster. The unique values of $\{z_1, \cdots, z_N\}$ induce partitioning (or clustering) of the set $\{1, \cdots, N\}$, and the distribution over the partitions is called the *Chinese restaurant process* (CRP) [138]. This equation (4.32) is equivalent to the expression of the CRP described in (4.28).

4.3.4 *Infinite Mixture Models*

In the previous section, Dirichlet processes and two practical representations, called the stick-breaking process and the Chinese restaurant process, were discussed. Dirichlet processes provide distributions over probability measures. The probability measure $\boldsymbol{G}$ sampled

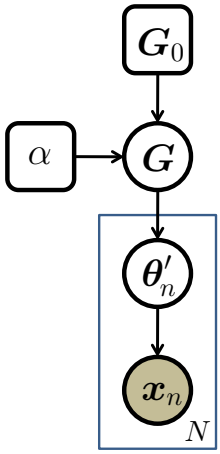


Figure 4.4: *The graphical model of the infinite mixture model. In the graphical model, the plate represents replication. The circle denotes variable, and the gray one denotes observation. The square represents fixed parameters. This type of representation of graphical models is used throughout this thesis.*

from a Dirichlet process is discrete. This makes Dirichlet processes too limited to model the continuous observations directly. In order to address this problem, a continuous density can be obtained by smoothing the random measure $\boldsymbol{G}$ with a density function. This results in an *infinite mixture model* [131, 149, 177].

According to the stick-breaking construction of the Dirichlet process (4.27), the random measure $\boldsymbol{G}$ sampled from this process can be specified by a probability mass function $P_G(\boldsymbol{\theta}) = \sum_{m=1}^{\infty} \pi_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)$. Given the random measure $\boldsymbol{G}$ which is specified by $P_G(\boldsymbol{\theta})$, and the density function of a single parametric model $p(\boldsymbol{x}|\boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$, the density of the *infinite mixture model* can be obtained by smoothing the random measure $\boldsymbol{G}$ with $p(\boldsymbol{x}|\boldsymbol{\theta})$ [177]:

$$p(\boldsymbol{x}|\mathcal{G}) = \int P_G(\boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{\theta} = \sum_{m=1}^{\infty} \pi_m p(\boldsymbol{x}|\boldsymbol{\theta}_m) \tag{4.33}$$

where $\mathcal{G} = \{\boldsymbol{\pi}, \boldsymbol{\Theta}\}$ are the parameters of the whole infinite mixture model, $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are the mixture weights, and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ are the parameters of all the components.

The graphical model of this infinite mixture model is illustrated in Figure 4.4, and the

corresponding generative process can be described as follows:

$$\begin{aligned} \boldsymbol{G} &\sim \mathrm{DP}(\alpha, \boldsymbol{G}_0) \\ \boldsymbol{\theta}'_n &\sim \boldsymbol{G} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}'_n) \end{aligned} \tag{4.34}$$

Since the random measure $\boldsymbol{G}$ is discrete, the parameters $\{\boldsymbol{\theta}'_1, \cdots, \boldsymbol{\theta}'_N\}$ from $\boldsymbol{G}$ have positive probability taking identical values.

From the two representations of the Dirichlet process, the stick-breaking process (4.26) and the Chinese restaurant process (4.28), the infinite mixture model can be described in two different but equivalent ways. The graphical model of the infinite mixture model based on the stick-breaking process is illustrated in the left plot of Figure 4.5. The corresponding generative process of the infinite mixture model can be described as:

$$\begin{aligned} \boldsymbol{\pi} &\sim \mathrm{GEM}(\alpha) \\ z_n &\sim \mathrm{Categorical}(\boldsymbol{\pi}) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{G}_0 \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \end{aligned} \tag{4.35}$$

where the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are given by the stick-breaking process (4.26), and z_n is the indicator variable that denotes with which component the nth observation is associated. Categorical$(\cdot)$ is the categorical distribution, which is the generalisation of the Bernoulli distribution with multiple possible outcomes (rather than two).

The graphical model of the infinite mixture model based on the Chinese restaurant process is illustrated in the right plot of Figure 4.5. The corresponding generative process of the infinite mixture model can be described as:

$$\begin{aligned} \boldsymbol{z} &\sim \mathrm{CRP}(\alpha) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{G}_0, \ \forall m \in \boldsymbol{z} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \end{aligned} \tag{4.36}$$

where the indicators $\boldsymbol{z} = \{z_1, \cdots, z_N\}$ are sampled from the Chinese restaurant process (4.28). In the infinite mixture model, typically, the density $p(\boldsymbol{x}|\boldsymbol{\theta})$ is an exponential family distribution [16], and the base distribution $\boldsymbol{G}_0$ is the conjugate prior for $p(\boldsymbol{x}|\boldsymbol{\theta})$. This

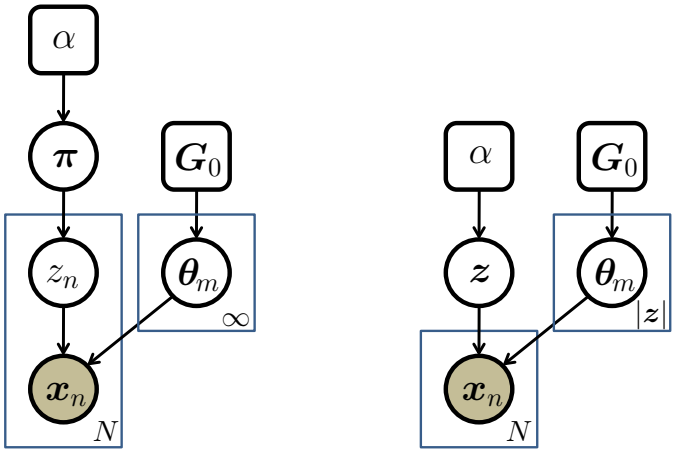


Figure 4.5: *The graphical models of the infinite mixture models based on different representations of the Dirichlet process. The left graphical model is based on the stick-breaking process, and the right model is based on the Chinese restaurant process. In the right plot, $|\boldsymbol{z}|$ denotes the number of unique values in set $\boldsymbol{z} = \{z_1, \cdots, z_N\}$.*

conjugacy leads to tractable posterior inference. When $\boldsymbol{G}_0$ is not the conjugate prior for $p(\boldsymbol{x}|\boldsymbol{\theta})$, efficiency inference also can also be made by using the sampling based scheme with auxiliary parameters introduced by Neal [131].

4.3.4.1 *Infinite Limit of Finite Mixture Models*

The previous section introduced infinite mixture models and two representations of the infinite mixture model based on the stick-breaking process and the Chinese restaurant process (illustrated in Figure 4.5). In this section, a different perspective on infinite mixture models will be discussed. Infinite mixture models can be viewed as the infinite limit of finite mixture models (when the number of components goes to infinity).

Consider a mixture model with M (finite) components, and each component with a density function $p(\boldsymbol{x}|\boldsymbol{\theta}_m)$. The mixture weights in the model are given by a symmetric Dirichlet distribution: Dirichlet$(\alpha/M, \cdots, \alpha/M)$. Then, the generative process of the

mixture model can be described as:

$$\begin{aligned}\boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha/M, \cdots, \alpha/M)\\ z_n &\sim \text{Categorical}(\boldsymbol{\pi})\\ \boldsymbol{\theta}_m &\sim p(\boldsymbol{\theta})\\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n})\end{aligned} \tag{4.37}$$

where $p(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$, and Categorical$(\cdot)$ is the categorical distribution. z_n is the indicator variable that denotes with which component the observation $\boldsymbol{x}_n$ is associated.

Given the mixture weights $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_M\}$, the joint probability of the indicator variables can be written as:

$$P(z_1, \cdots, z_N|\pi_1, \cdots, \pi_M) = \prod_{m=1}^{M} \pi_m^{N_m} \tag{4.38}$$

where N_m is the number of data associated with the mth component, namely $N_m = \sum_{n=1}^{N} \delta(z_n, m)$, and $\delta(\cdot)$ is the Kronecker delta. By marginalising out the mixture weights $\boldsymbol{\pi}$, which are given by a Dirichlet distribution described in (4.37), the joint probability of the indicator variables (4.38) can be further written as:

$$\begin{aligned}P(z_1, \cdots, z_N|\alpha) =& \int P(z_1, \cdots, z_N|\pi_1, \cdots, \pi_M)p(\pi_1, \cdots, \pi_M|\alpha)d(\pi_1, \cdots, \pi_M)\\ =& \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \prod_{m=1}^{M} \frac{\Gamma(N_m + \alpha/M)}{\Gamma(\alpha/M)}\end{aligned} \tag{4.39}$$

Given the joint distribution of the indicators (4.39), the conditional distribution of indicator z_N given other $N-1$ indicators $\{z_1, \cdots, z_{N-1}\}$ can be described as follows:

$$\begin{aligned}P(z_N = m|z_1, \cdots, z_{N-1}, \alpha) &= P(z_N|z_1, \cdots, z_{N-1}, \alpha)|_{z_N=m}\\ &= \frac{P(z_1, \cdots, z_N|\alpha)}{P(z_1, \cdots, z_{N-1}|\alpha)}\bigg|_{z_N=m} = \frac{N_m + \alpha/M}{N-1+\alpha}\end{aligned} \tag{4.40}$$

So far, M is considered as a finite number. When the number of components goes to infinite, $M \to \infty$, mixture models with infinite number of components can be resulted. Since the number of components becomes infinite, the components can be divided up into two groups: the *represented components* that have associated data and the *unrepresented*

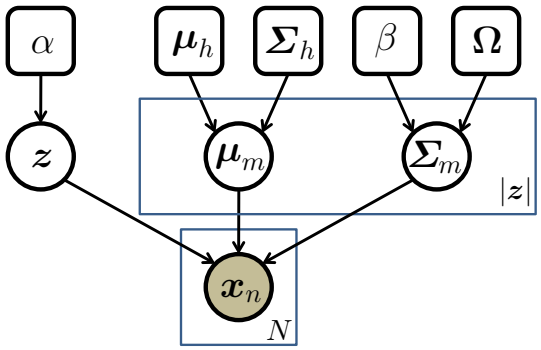


Figure 4.6: *The graphical model of the infinite Gaussian mixture model.* $|\boldsymbol{z}|$ *denotes the number of unique values in set* $\boldsymbol{z} = \{z_1, \cdots, z_N\}$.

components that have no associated data. Thus, when $M \to \infty$, the conditional distribution of the indicator $P(z_N = m|z_1, \cdots, z_{N-1}, \alpha)$ described in equation (4.40) can be further described as [149]:

$$P(z_N = m|z_1, \cdots, z_{N-1}, \alpha) = \begin{cases} \dfrac{N_m}{N-1+\alpha}, & \text{when } m \text{ is a represented component} \\ \dfrac{\alpha}{N-1+\alpha}, & \text{when } m \text{ is an unrepresented component} \end{cases} \tag{4.41}$$

This is the Chinese restaurant process described in (4.28). Thus, when the number of components M goes to infinite, the sequence of indicators $\boldsymbol{z} = \{z_1, \cdots, z_N\}$ is obtained from the Chinese restaurant process, namely $\boldsymbol{z} \sim \text{CRP}(\alpha)$[1]. Then the generative process of mixture models with infinite number of components can be described as:

$$\begin{aligned} \boldsymbol{z} &\sim \text{CRP}(\alpha) \\ \boldsymbol{\theta}_m &\sim p(\boldsymbol{\theta}), \ \forall m \in \boldsymbol{z} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \end{aligned} \tag{4.42}$$

This is the infinite mixture model based on the Chinese restaurant process described in (4.36) with base distribution $\boldsymbol{G}_0 = p(\boldsymbol{\theta})$. Thus, infinite mixture models are the limit of finite mixture models (when the number of components goes to infinite $M \to \infty$).

[1] Given the exchangeability of the indicator variables, each indicator can be sampled (as the last one with index N) given all other $N-1$ indicators.

4.3.4.2 Relationships with Infinite GMMs

The *infinite Gaussian mixture model* (iGMM) was first introduced by Rasmussen [149]. The infinite GMM is a infinite mixture model with Gaussian components. For each Gaussian component of the infinite GMM, the mean $\boldsymbol{\mu}_m$ and covariance $\boldsymbol{\Sigma}_m$ are assumed to be independent of each other, and each gives a conjugate prior (the Gaussian distribution and the inverse Wishart distribution respectively) [77, 149]. The graphical model of the infinite GMM is illustrated in Figure 4.6, and the corresponding generative process can be described as:

$$\begin{aligned} \boldsymbol{z} &\sim \text{CRP}(\alpha) \\ \boldsymbol{\mu}_m &\sim \mathcal{N}(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h), \ \forall m \in \boldsymbol{z} \\ \boldsymbol{\Sigma}_m &\sim \text{Wishart}^{-1}(\beta, \boldsymbol{\Omega}), \ \forall m \in \boldsymbol{z} \\ \boldsymbol{x}_n &\sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n}) \end{aligned} \tag{4.43}$$

where $\mathcal{N}(\cdot)$ is a Gaussian distribution, and $\text{Wishart}^{-1}(\cdot)$ is a inverse Wishart distribution. Compared with the infinite mixture model described in (4.36), the base distribution for the infinite GMM is $\boldsymbol{G}_0 = p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\mu})p(\boldsymbol{\Sigma})$.

4.3.5 Infinite Mixtures of Experts

In the previous section, infinite mixture models based on Dirichlet processes were discussed. This type of model is generative, which models the density of the data. For a classification task, discriminative models might be preferred [105, 132]. In this section, a type of mixture of discriminative models based on the Dirichlet process called the *infinite mixture of experts* will be studied.

4.3.5.1 Mixtures of Experts

Rather than making predictions using a single classifier (a conditional distribution), which might be inadequate to model the whole training data, it is possible to choose different conditional distributions to make predictions according to different inputs. If the choice is input independent, the resulting model is called the *conditional mixture model* [16]. If the

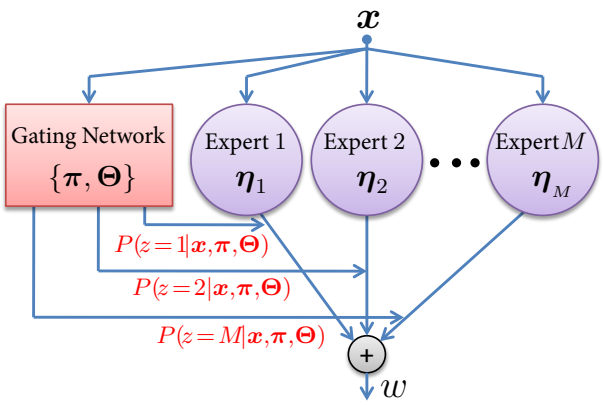


Figure 4.7: *The framework of the mixture of experts.*

choice of the conditional distributions is input dependent, when the choice is a hard decision, this framework is known as a *decision tree* [16]; when the choice is given a probability depending on the input, this is known as a *mixture of experts* [16, 91, 99].

The framework for mixture of experts with M experts is illustrated in Figure 4.7. As shown in this figure, the weight $P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta})$ for each expert is input dependent and determined by the *gating network*. If the gating network is based on a Gaussian mixture model (GMM), then the probability $P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta})$ is the component posterior probability of the GMM[1]:

$$P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}) = \frac{\pi_z \mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}_z)}{\sum_z \pi_z \mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}_z)}, \quad z \in \{1, 2, \ldots, M\} \tag{4.44}$$

where M is the number of experts, and z is the indicator variable, that denotes which expert the input $\boldsymbol{x}$ is associated with. $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_M\}$ are the mixture weights and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are the model parameters of the components. $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}_z)$ is the zth Gaussian component with parameters $\boldsymbol{\theta}_z$.

The zth *expert* is a classifier[2] having conditional probability $P(w|\boldsymbol{x}, \boldsymbol{\eta}_z)$ with parameters $\boldsymbol{\eta}_z$, where w is the class label. Given an input $\boldsymbol{x}$, the overall conditional distribution of

[1] Gating networks of (infinite) mixtures of experts are assumed to be (infinite) mixture models throughout this thesis. More generally, any suitable input dependent function can be the gating network, e.g. a softmax function adopted in [99] and the function based on a Dirichlet process and Gaussian kernel functions in [150].

[2] Experts are assumed to be classifiers throughout this thesis. Alternatively, they can also be regression models [16].

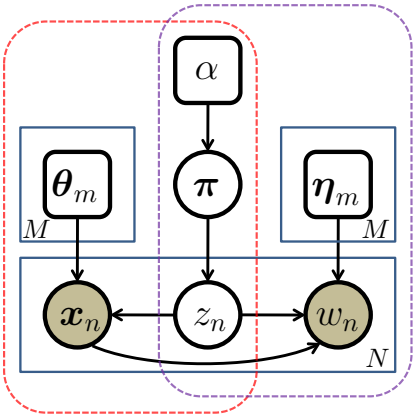


Figure 4.8: *The graphical model of the mixture of experts. The plot associated with the red dotted line represents the gating network, and the plot associated with the purple dotted line represents the experts.*

the class w for the mixture of experts with M experts can be described as:

$$P(w|\boldsymbol{x}, \mathcal{G}) = P(w|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}) = \sum_z P(w|\boldsymbol{x}, \boldsymbol{\eta}_z) P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}), \quad z \in \{1, 2, \cdots, M\} \tag{4.45}$$

where $\mathcal{G} = \{\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}\}$ are the parameters of the whole model, and $\boldsymbol{H} = \{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_M\}$ are the parameters of all the experts. $P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta})$ is the gating network described in (4.44).

When the gating network is given by the component posteriors of a Gaussian mixture model and the number of experts is M, the graphical model of the mixture of experts is illustrated in Figure 4.8, and the corresponding generative process of this model can be described as follows:

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha/M, \cdots, \alpha/M) \\ z_n &\sim \text{Categorical}(\boldsymbol{\pi}) \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \\ w_n &\sim P(w|\boldsymbol{x}_n, \boldsymbol{\eta}_{z_n}) \end{aligned} \tag{4.46}$$

where the mixture weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_M\}$ are are drawn from a *symmetric Dirichlet distribution* [16] with concentration parameter α, and z_n is the indicator variable that denotes the nth observation is associated with which expert. Categorical$(\cdot)$ is the *categorical distribution* which is the generalisation of the *Bernoulli distribution* with M possible outcomes.

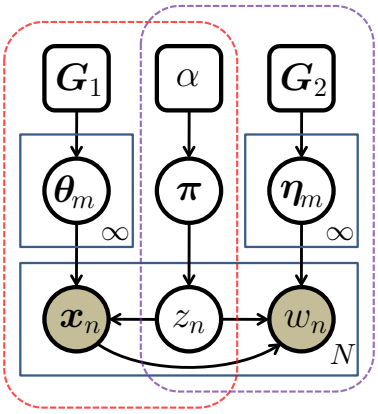


Figure 4.9: *The graphical model of the infinite mixture of experts based on the stick-breaking process. The plot associated with red dotted line is the stick-breaking construction of the infinite mixture model.*

$p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n})$ is the density function for z_nth component, and $P(w|\boldsymbol{x}_n, \boldsymbol{\eta}_{z_n})$ is the conditional distribution for the z_nth expert.

4.3.5.2 *Infinite Limit*

As a parametric model, the number of experts M in the mixture of experts needs to be given in advance. In order to bypass the problem of choosing model complexity, a Bayesian non-parametric version of the mixture of experts called an infinite mixture of experts will be studied. As discussed in section 4.3.4.1, an infinite mixture model is the infinite limit of a finite mixture model (when the number of components goes to infinite). In a similar fashion, the non-parametric counterpart of the mixture of experts called the *infinite mixture of experts* can be derived, when the number of experts goes to infinity in the mixture of experts, namely $M \to \infty$. Given an input $\boldsymbol{x}$, the overall conditional distribution of the class w described in (4.45) becomes summing over infinite number of experts:

$$P(w|\boldsymbol{x}, \mathcal{G}) = P(w|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}) = \sum_z P(w|\boldsymbol{x}, \boldsymbol{\eta}_z) P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \quad z \in \{1, 2, \cdots, \infty\} \tag{4.47}$$

where $\mathcal{G} = \{\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}\}$ are the parameters of the whole model, $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are the mixture weights, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ are the parameters of all the components, and $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of all the experts.

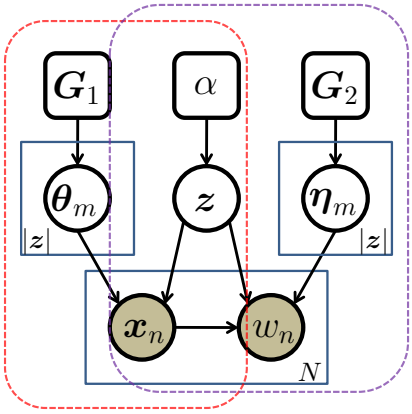


Figure 4.10: *The graphical model of the infinite mixture of experts based on the Chinese restaurant process (CRP). The plot associated with red dotted line is the CRP construction of the infinite mixture model.*

In section 4.3.4 two practical constructions of infinite mixture models were discussed, namely the infinite mixture models based on the stick-breaking process and the Chinese restaurant process. Infinite mixtures of experts can also be described by these two processes. The graphical model of the infinite mixture of experts based on the stick-breaking process is illustrated in Figure 4.9. The corresponding generative process of the infinite mixture of experts can be described as:

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{GEM}(\alpha) \\ z_n &\sim \text{Categorical}(\boldsymbol{\pi}) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{G}_1, \ \boldsymbol{\eta}_m \sim \boldsymbol{G}_2 \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \\ w_n &\sim P(w|\boldsymbol{x}_n, \boldsymbol{\eta}_{z_n}) \end{aligned} \tag{4.48}$$

where the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are given by the stick-breaking process (4.26), and z_n is the indicator variable that denotes with which expert the nth observation is associated. Categorical$(\cdot)$ is the categorical distribution. $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are base distributions of the Dirichlet process, namely $\boldsymbol{G}_0 = \boldsymbol{G}_1\boldsymbol{G}_2$. $p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n})$ is the density function of the z_nth component, and $P(w|\boldsymbol{x}_n, \boldsymbol{\eta}_{z_n})$ is the conditional distribution of the class w given by the z_nth expert.

In addition to the stick-breaking process, the Chinese restaurant process provides a mechanism to draw samples from the Dirichlet process without specifying the underlying distribution. Based on the Chinese restaurant process, the graphical model of the infinite mixture of experts is illustrated in Figure 4.10. The corresponding generative process of the infinite mixture of experts can be described as:

$$\begin{aligned} \boldsymbol{z} &\sim \text{CRP}(\alpha) \\ \boldsymbol{\theta}_m \sim \boldsymbol{G}_1,\ \boldsymbol{\eta}_m &\sim \boldsymbol{G}_2,\ \forall m \in \boldsymbol{z} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \\ w_n &\sim P(w|\boldsymbol{x}_n, \boldsymbol{\eta}_{z_n}) \end{aligned} \tag{4.49}$$

where the indicators $\boldsymbol{z} = \{z_1, \cdots, z_N\}$ corresponding to observations $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ are given by the Chinese restaurant process with concentration parameter α described in (4.28).

To summarise, the infinite mixture of experts defines an infinite mixture of conditional models (experts). For the infinite mixture of experts, each expert is a discriminative model, and the gating network is given by an infinite mixture model based on the Dirichlet process, which is a generative model. For discriminative models, the conditional distribution of the class w is modelled directly without considering the underlying distribution of the observations. In contrast, the infinite mixture of experts gives the conditional distribution of the class, but the underlying distribution of the observations also is modelled by the gating network. Thus, the infinite mixture of expert discussed in this section is a combination of generative models (gating network) and discriminative models (experts).

4.3.5.3 *Relationships with Infinite SVMs*

The *infinite support vector machine*[1], first introduced by Zhu [213], is an example of the infinite mixture of experts discussed in the previous section. For the infinite SVM, the gating network is given by an infinite mixture model based on the stick-breaking process discussed in section 4.3.4, and each expert is a maximum entropy discrimination (MED) [90, 93] large margin classifier (which is discussed in Appendix B.1.1). Thus, the infinite SVM can be described as an infinite mixture of experts based on the stick-breaking process illustrated in Figure 4.9 [213].

[1] The infinite support vector machine is discussed in detail in Appendix B.

For the infinite mixture of experts described in section 4.3.5.2, each expert is a discriminative model, which gives the conditional distribution of the class label w given input $\boldsymbol{x}$. In contrast, for the infinite SVM, each expert is a discriminant function, which gives a mapping from the input $\boldsymbol{x}$ to the class label w. Analogous to the overall conditional distribution given by the infinite mixture of experts described in equation (4.47), the overall discriminant function for the infinite SVM can be described as [213]:

$$F(w, \boldsymbol{x}; \mathcal{G}) = F(w, \boldsymbol{x}; \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}) = \sum_z F(w, \boldsymbol{x}; \boldsymbol{\eta}_z) P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \quad z \in \{1, 2, \cdots, \infty\} \tag{4.50}$$

where $F(w, \boldsymbol{x}; \boldsymbol{\eta}_z)$ is the discriminant function for the zth expert. the weight from the gating network $P(z|\boldsymbol{x}, \boldsymbol{\pi}, \boldsymbol{\Theta})$ is defined in (4.44) with component number $M = \infty$, which is the component posterior of the infinite mixture model. The infinite SVM is discussed in detail in appendix B. The commonly used processes such as hierarchical Dirichlet processes and beta processes are also discussed in appendices C and D.

4.4 Some Applications in Speech Processing

Bayesian non-parametric models have been widely and successfully employed in various areas [28, 50, 168]. Since this thesis focuses on speech recognition, examples of non-parametric models used in speech processing will be discussed in this section.

4.4.1 *Topic Modelling*

Topic modelling has been widely used in the field of information retrieval [35, 206]. One of the most successful examples is Internet search engines. Topic modelling is employed to model the text corpora and other collections of discrete data, and the purpose is to find short descriptions of the members of a collection that enable efficient processing of large collections [21].

When modelling the word occurrences in a set of documents, one simple solution is to place each document with a single topic. However, it is more appropriate to allow each document to contain more than one topics. For instance, the travel book might contain topics about weather, travel information, tourist destinations and history. Thus, it would be

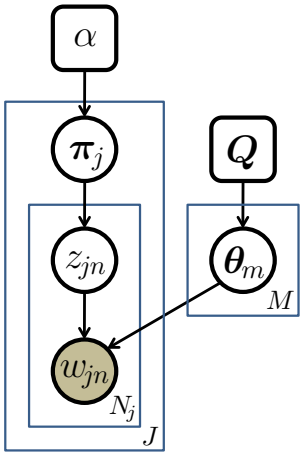


Figure 4.11: *The graphical model of latent Dirichlet allocation.*

more appropriate to assign several topics to one document. Assume there are M topics, and the probability of choosing one topic is given by the topic proportion $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_M\}$:

$$z \sim \text{Categorical}(\boldsymbol{\pi}) \tag{4.51}$$

where the indicator variable z denotes which topic is chosen, and Categorical$(\cdot)$ is the Categorical distribution, which is the generalisation of the Bernoulli distribution with multiple possible outcomes. Given the indicator variables, a word in this document then can be considered as being generated from the chosen topic. The process of generating the words for each document can described as the follows:

$$\begin{aligned} z_n &\sim \text{Categorical}(\boldsymbol{\pi}) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ w_n &\sim P(w|\boldsymbol{\theta}_{z_n}) \end{aligned} \tag{4.52}$$

where $P(w|\boldsymbol{\theta}_{z_n})$ is the words distribution given the topic z_n with parameters $\boldsymbol{\theta}_{z_n}$, and it is typically a multinomial distribution. The parameters $\boldsymbol{\theta}_m$ are placed with a prior distribution $\boldsymbol{Q}$.

In latent Dirichlet allocation (LDA) [21], the topic proportion $\boldsymbol{\pi}_j$ is given a symmetric Dirichlet distribution. Assume there are J documents, the generative process for LDA can

be described as the follows:

$$\begin{aligned} \boldsymbol{\pi}_j &\sim \text{Dirichlet}(\alpha) \\ z_{jn} &\sim \text{Categorical}(\boldsymbol{\pi}_j) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ w_{jn} &\sim P(w|\boldsymbol{\theta}_{z_{jn}}) \end{aligned} \tag{4.53}$$

where the parameters $\boldsymbol{\theta}_m$ are distributed according to a Dirichlet distribution $\boldsymbol{Q}$. Alternatively, $\boldsymbol{\theta}_m$ can be treated as unknown but fixed parameters, and estimated in an empirical Bayesian fashion as described in (4.9). $P(w|\boldsymbol{\theta}_{z_{ji}})$ is a multinomial distribution. The corresponding graphical model of LDA is illustrated in Figure 4.11.

In LDA the words for each documents are given by a mixture of topics, and these topics are shared among documents. Thus, LDA extends standard mixture models by sharing a common set of components among different related groups. In LDA, the number of topics is set to a fixed value M. In order to circumvent the problem of setting model complexity, a Bayesian non-parametric model can be employed. In LDA different documents share a common set of topics. The hierarchical Dirichlet process (HDP)[1] provides a mechanism to link groups of data by sharing components. Thus, by using the framework of the HDP, LDA can be extended to be a Bayesian non-parametric model called *HDP-LDA* [175]:

$$\begin{aligned} \boldsymbol{c} &\sim \text{GEM}(\beta) \\ \boldsymbol{\pi}_j &\sim \text{DP}(\alpha, \boldsymbol{c}) \\ z_{jn} &\sim \text{Categorical}(\boldsymbol{\pi}_j) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ w_{jn} &\sim P(w|\boldsymbol{\theta}_{z_{jn}}) \end{aligned} \tag{4.54}$$

where $\text{GEM}(\cdot)$ is the stick-breaking process described in (4.26), and $P(w|\boldsymbol{\theta}_{z_{jn}})$ is the distribution over words for the z_{jn}th topic. The corresponding graphical model of HDP-LDA is illustrated in Figure 4.12. In this graphical model, J is the number of documents, N_j is the number of words in the jth document. In addition to use the framework of the HDP, extending LDA by employing a *nested Chinese restaurant process* is discussed in [20].

[1] The hierarchical Dirichlet process is discussed in detail in Appendix C.

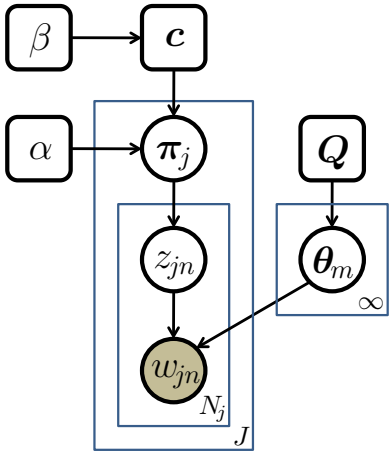


Figure 4.12: *The graphical model of HDP-LDA.*

4.4.2 Word Segmentation

Word segmentation is the problem of discovering word boundaries in continuous speech or text. This problem is non-trivial, since some languages do not have explicit word boundaries, such as Chinese and Japanese [80, 114]. In [75] statistical approaches are proposed for word segmentation: the Dirichlet process is employed in the unigram modelling assumption of word dependencies, and the hierarchical Dirichlet process in the bigram assumption.

Under the unigram modelling assumption, the words in an utterance are independent from each other. Thus word sequence can be generated according to the following process:

$$\begin{aligned} \boldsymbol{G} &\sim \mathrm{DP}(\alpha, \boldsymbol{Q}) \\ w_n &\sim \boldsymbol{G} \end{aligned} \tag{4.55}$$

where the base distribution $\boldsymbol{Q}$ can be described as:

$$\boldsymbol{Q} = \sum_{m=1}^{\infty} \pi_m \delta(w, w_m) \tag{4.56}$$

$$\pi_m = P(w_m) = P_0(1-P_0)^{l-1} \prod_{i=1}^{l} P(m_i) \tag{4.57}$$

where the word w_m is composed of the phones $w_m = \{m_1, \cdots, m_l\}$, $P(m_i)$ is the probability of phone m_i, and P_0 is the probability of the word boundary. In [75] utterance boundaries are given, and each utterance has been converted to a phonemic representation using a phonemic dictionary. Thus each utterance to be segmented is a sequence of phones.

Under the bigram modelling assumption, the probability of generating the current word w_n is only dependent on the previous word w_{n-1}. Thus, a hidden Markov model (HMM) can be employed to model the dependencies of words, where each latent state of the HMM corresponds to a word. Since there are an unbounded number of potential words, it would be more appropriate to have infinite possible number of states in the HMM. Thus the framework of the hierarchical Dirichlet process (HDP) can be applied. The word sequence then can be generated according to a HDP-HMM:

$$\begin{aligned} \boldsymbol{G}_0 &\sim \mathrm{DP}(\beta, \boldsymbol{Q}) \\ \boldsymbol{G}_w &\sim \mathrm{DP}(\alpha, \boldsymbol{G}_0) \\ w_i | w_{i-1} &\sim \boldsymbol{G}_{w_{i-1}} \end{aligned} \tag{4.58}$$

This is the direct description of the HDP-HMM. An alternative stick-breaking construction of the HDP-HMM is described in (C.13), and the corresponding graphical model is illustrated in Figure C.5. As demonstrated in [75], significant improvements in segmentation accuracy can be achieved by employing this HDP-HMM approach.

4.4.3 *Speaker Diarisation*

Speaker diarisation is the problem of segmenting an audio recording into time intervals corresponding to individual speakers and assigning speaker labels [50, 195]. Speaker diarisation is a joint problem of segmenting and clustering, this makes the HMM a suitable model in which the model transitions among states which are associated with different speakers [50]. In standard HMMs, the number of states needs to be set in advance. If each state corresponds to a speaker, the number of speakers must be given. This is limited, since normally there is no priori knowledge on the number of the speakers in a meeting. Thus, in a speaker diarisation problem, it would be more appropriate to employ the framework of hierarchical Dirichlet processes (HDP), which allows infinite possible number of states (speakers).

As discussed in [50, 51], the HDP-HMM is often inadequate to model temporal persistence of the states, as the HDP-HMM tends to rapidly switch between redundant states. This leads to a poor performance in speaker diarisation [51]. To solve this problem, the *sticky HDP-HMM* [49, 50] was proposed to increase the prior probability of self-transitions. The

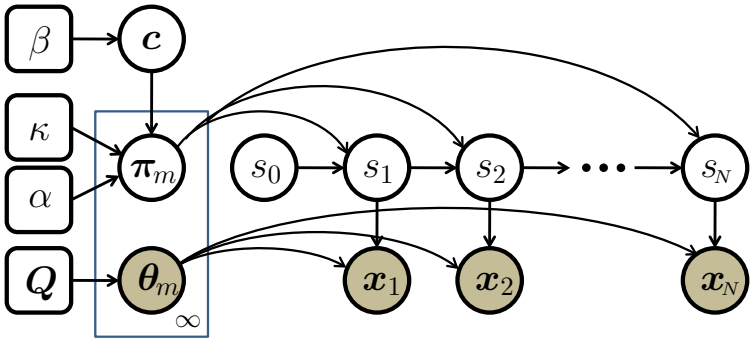


Figure 4.13: *The graphical model of the sticky HDP-HMM.*

model can jointly segment and cluster the audio into speaker homogenous regions with better performance. The graphical model of the sticky HDP-HMM is illustrated in Figure 4.13, and the corresponding generative process can be described as the follows:

$$\begin{aligned} \boldsymbol{c} &\sim \text{GEM}(\beta) \\ \boldsymbol{\pi}_m &\sim \text{DP}\Big(\alpha+\kappa, \frac{\alpha\boldsymbol{c}+\kappa\delta(m,m)}{\alpha+\kappa}\Big) \\ s_n|s_{n-1} &\sim \text{Categorical}(\boldsymbol{\pi}_{s_{n-1}}) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{s_n}) \end{aligned} \tag{4.59}$$

where the weights $\boldsymbol{c} = \{c_1, c_2, \ldots\}$ are generated according to the stick-breaking process described in (4.26), and s_n is the state indicator. Given the state indicator s_n, the observation $\boldsymbol{x}_n$ is distributed according to $p(\boldsymbol{x}|\boldsymbol{\theta}_{s_n})$ which is a Gaussian distribution, and $\boldsymbol{\theta}_{s_n}$ is generated according to the base distribution $\boldsymbol{Q}$. Compared with the standard HDP-HMM described in (C.13) of Appendix C, an extra parameter called the self-transition bias parameter κ is introduced in the sticky HDP-HMM. By introducing κ, the expected probability of self-transition can be increased [50].

In speaker diarisation, since each speaker is associated with a state of the HMM. Generally, the HMM with high performance employs Gaussian mixture models (GMMs) as the distributions of the emitting states [200]. Thus, it would be more appropriate to use a mixture model as the emitting state distribution in the sticky HDP-HMM. Then each speaker corresponds to a mixture model, more specifically a Dirichlet process mixture model which gives all possibles of distributions. The graphical model of the sticky HDP-HMM with

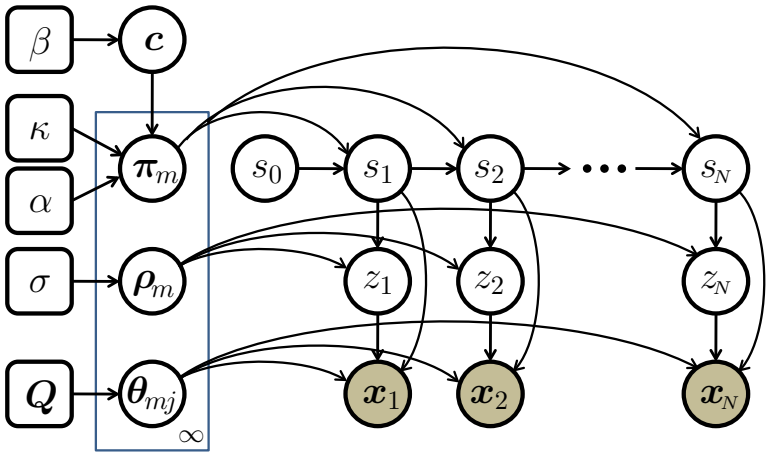


Figure 4.14: *The graphical model of the sticky HDP-HMM with Dirichlet process mixture model emissions.*

Dirichlet process mixture model emissions is illustrated in Figure 4.14, and the corresponding generative process can be described as follows [49, 50]:

$$
\begin{aligned}
\boldsymbol{c} &\sim \text{GEM}(\beta) \\
\boldsymbol{\pi}_m &\sim \text{DP}\Big(\alpha + \kappa, \frac{\alpha \boldsymbol{c} + \kappa \delta(m, m)}{\alpha + \kappa}\Big) \\
s_n | s_{n-1} &\sim \text{Categorical}(\boldsymbol{\pi}_{s_{n-1}}) \\
\boldsymbol{\rho}_m &\sim \text{GEM}(\sigma) \\
z_n &\sim \boldsymbol{\rho}_{s_n} \\
\boldsymbol{\theta}_{mj} &\sim \boldsymbol{Q} \\
\boldsymbol{x}_n &\sim p(\boldsymbol{x} | \boldsymbol{\theta}_{s_n z_n})
\end{aligned}
\tag{4.60}
$$

where the weights $\boldsymbol{c} = \{c_1, c_2, \ldots\}$ and $\boldsymbol{\rho}_m = \{\rho_{m1}, \rho_{m2}, \ldots\}$ are generated according to the stick-breaking process described in (4.26). Again s_n is the state indicator, and each state of the HDP-HMM is given a DP mixture model. The component indicator z_n is introduced to denote with which component the observation is associated. Given the state indicator s_n and the component indicator z_n, the observation $\boldsymbol{x}_n$ is distributed according to $p(\boldsymbol{x} | \boldsymbol{\theta}_{s_n z_n})$ which is a Gaussian distribution, and $\boldsymbol{\theta}_{s_n z_n}$ is generated according to the base distribution $\boldsymbol{Q}$. By using this type of sticky HDP-HMMs with special treatments of self-transitions, the state-of-the-art diarisation performance can be achieved [50].

4.5 Summary

In this chapter, motivations of Bayesian non-parametric models and de Finetti's theorem were discussed in section 4.1. The basic ideas of Bayesian inference and conditional Bayesian inference were introduced in section 4.2. Section 4.3 introduced the Dirichlet process, and two representations of this process called the Chinese restaurant process and the stick-breaking process. The frameworks of mixture models and mixture of experts based on Dirichlet processes were discussed in detail in section 4.3.4 and 4.3.5 respectively. The infinite Gaussian mixture model and infinite support vector machine can be subsumed under these two frameworks. Finally, some applications of the Bayesian non-parametric models in speech processing were briefly discussed in section 4.4.

Chapter 5

Infinite Structured Discriminative Models

In Chapter 4, Bayesian inference was introduced and some of the commonly used Bayesian non-parametric models were briefly discussed. In this chapter, a criterion-based perspective on Bayesian inference will be introduced. Here Bayesian inference is interpreted as a minimisation criterion consisting of two terms: one representing the prior beliefs; and a second representing information from the observations. Furthermore, this minimisation criterion can be subsumed under a general criterion (with a log-likelihood criterion function). This general criterion allows different forms of criterion functions to be used. Finally, training of the infinite structured discriminative models using the general criterion with different criterion functions will be discussed in detail in this chapter.

5.1 Criterion-based Perspectives on Bayesian Inference

As discussed in section 4.2, Bayesian inference is the analysis of beliefs. Before any data is observed, the prior distribution over model parameters is used to express the available knowledge (or prior beliefs) on the data generation process; and the beliefs can be updated when observing new evidence. The most natural way of combining new evidence with prior beliefs is the application of Bayes' rule. There are two factors in Bayesian inference: prior beliefs before observing data and updating beliefs when observing new evidence. In section 4.2, the posterior distribution of the model parameters obtained through Bayes' rule was

discussed. In [93, 214] an alternative posterior distribution (which differs from the posterior distribution obtained through Bayes' rule) is interpreted as being obtained from a criterion. As described by Zellner [205], the posterior distribution obtained through Bayes' rule is equivalent to a distribution obtained from a minimisation criterion (which consists of two terms: one term represents the prior beliefs and another term represents information from the observations). This minimisation criterion can be subsumed under a general criterion with a criterion function consisting of log-likelihoods. With different forms of criterion functions, this general criterion can result in different meaningful criteria, e.g. the large margin training criterion. This type of criterion will be discussed in detail in the rest of this section.

For a generative model, the posterior distribution of the model parameters $\mathcal{G}$ can be described as minimising the Kullback-Leibler (KL) divergence $\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G}|\mathcal{D})\big)$ between the distribution $q(\mathcal{G})$ to be estimated and the posterior distribution $p(\mathcal{G}|\mathcal{D})$ obtained by Bayes' rule [205]: $p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G})p(\mathcal{D}|\mathcal{G})$. Let $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be training data, $p(\mathcal{D}|\mathcal{G}) = \prod_n p(\boldsymbol{x}_n|\mathcal{G})$ the likelihood, and $p(\mathcal{G})$ the prior distribution. When the estimated distribution $q(\mathcal{G})$ equals the posterior distribution $p(\mathcal{G}|\mathcal{D})$ obtained by Bayes' rule, the KL divergence $\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G}|\mathcal{D})\big)$ by definition is zero and minimised. According to the definition of the KL divergence, the following expression can be derived:

$$\begin{aligned}\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G}|\mathcal{D})\big) &= \int q(\mathcal{G}) \log \frac{q(\mathcal{G})}{p(\mathcal{G}|\mathcal{D})} d\mathcal{G} \\ &= \int q(\mathcal{G}) \log \frac{q(\mathcal{G})}{p(\mathcal{G})} d\mathcal{G} - \int q(\mathcal{G}) \log p(\mathcal{D}|\mathcal{G}) d\mathcal{G} + \log p(\mathcal{D}) \qquad (5.1)\end{aligned}$$

Since the log probability of the training data $\log p(\mathcal{D})$ is not a function of $q(\mathcal{G})$, minimising the KL divergence $\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G}|\mathcal{D})\big)$ is equivalent to the following expression:

$$\begin{aligned}&\arg\min_{q(\mathcal{G})} \Big\{ \underbrace{\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G})\big)}_{\text{prior}} - \underbrace{\int q(\mathcal{G}) \log p(\mathcal{D}|\mathcal{G}) d\mathcal{G}}_{\text{evidence}} \Big\} \qquad (5.2) \\ &\text{s.t.} \quad q(\mathcal{G}) \in \mathcal{P}_{\text{prob}}\end{aligned}$$

where $\mathcal{P}_{\text{prob}}$ is a set consisting of all possible valid distributions over $\mathcal{G}$. The criterion described in (5.2) can be related to standard Bayesian inference. The prior beliefs are given by the first term. When new evidence (given by the second term in the form of the log-likelihood) is observed, the beliefs can be updated in the form of the posterior distribution

by minimising the criterion (5.2). Since this criterion is equivalent to minimise the KL divergence $\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G}|\mathcal{D})\big)$, the solution that minimise criterion (5.2) is:

$$\hat{q}(\mathcal{G}) = p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G})p(\mathcal{D}|\mathcal{G}) = p(\mathcal{G}) \prod_n p(\boldsymbol{x}_n|\mathcal{G}) \tag{5.3}$$

Thus, the posterior distribution obtained by minimising criterion (5.2) is equivalent to the posterior distribution through Bayes' rule described in (4.8) in section 4.2.1.

A similar approach can be adopted for discriminative models, where the conditional distribution of a class w is of interest. For a discriminative model $P(w|\boldsymbol{x}, \mathcal{G})$ with model parameters $\mathcal{G}$, given the training data $\mathcal{D} = \{\mathcal{X}, \mathcal{W}\} = \{(\boldsymbol{x}_1, w_1), \ldots, (\boldsymbol{x}_N, w_N)\}$, it is possible to formulate a criterion similar to that for a generative model described in (5.2). The posterior distribution $\hat{q}(\mathcal{G})$ can be estimated by minimising this criterion:

$$\begin{aligned} &\arg\min_{q(\mathcal{G})} \Big\{ \underbrace{\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G})\big)}_{\text{prior}} - \underbrace{\int q(\mathcal{G}) \log P(\mathcal{W}|\mathcal{X}, \mathcal{G}) d\mathcal{G}}_{\text{evidence}} \Big\} \\ &\text{s.t.} \quad q(\mathcal{G}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{5.4}$$

where $p(\mathcal{G})$ is the prior distribution, $P(\mathcal{W}|\mathcal{X}, \mathcal{G}) = \prod_n P(w_n|\boldsymbol{x}_n, \mathcal{G})$ is the conditional likelihood, and $\mathcal{P}_{\text{prob}}$ is the set of all possible valid distribution over $\mathcal{G}$. Analogous with (5.3), the solution that minimise criterion (5.4) is:

$$\hat{q}(\mathcal{G}) = p(\mathcal{G}|\mathcal{D}) = p(\mathcal{G}|\mathcal{X}, \mathcal{W}) \propto p(\mathcal{G})P(\mathcal{W}|\mathcal{X}, \mathcal{G}) = p(\mathcal{G}) \prod_n P(w_n|\boldsymbol{x}_n, \mathcal{G}) \tag{5.5}$$

This is the same as the posterior distribution obtained through Bayes' rule described in (4.15) in section 4.2.2.

5.2 The General Criterion

As discussed in section 5.1, the application of Bayes' rule to obtain the posterior distribution is equivalent to the minimisation criteria described in (5.2) and (5.4), where prior beliefs and evidence are embodied by the first and second term respectively. More generally, the optimal distribution can be viewed as a distribution obtained through a minimisation criterion

which consists of prior beliefs and evidence:

$$\arg\min_{q(\mathcal{G})} \Big\{ \underbrace{\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G})\big)}_{\text{prior}} - \underbrace{\int q(\mathcal{G})\mathcal{F}(\mathcal{G};\mathcal{D})d\mathcal{G}}_{\text{evidence}} \Big\} \tag{5.6}$$
$$\text{s.t.} \quad q(\mathcal{G}) \in \mathcal{P}_{\text{prob}}$$

where $\mathcal{F}(\mathcal{G};\mathcal{D})$ is the *criterion function*. This function must be real valued and represents the information from evidence. Moreover, the selection of the criterion function should guarantee to yield a valid distribution $q(\mathcal{G})$. For example, the integral over $\exp\big(\mathcal{F}(\mathcal{G};\mathcal{D})\big)$ should be finite, that ensures the normalisation term of $q(\mathcal{G})$ is finite, which will be discussed in section 5.2.1. Thus, the selection of the criterion function $\mathcal{F}(\mathcal{G};\mathcal{D})$ is highly constrained. The Gibbs infinite SVM [207] is a specification of this general criterion, where the criterion function consists of the hinge loss function and log-likelihoods. This general criterion is also related to regularised Bayesian [214] and the maximum entropy discrimination (MED) [90, 93], which will be discussed later. In this thesis, only three forms of the criterion function will be investigated, i.e. the hinge loss function, log-likelihoods and the combination of these two.

The minimisation criteria (5.2) and (5.4) are specifications of this general criterion (5.6), where the criterion functions are the logarithms of the likelihood and the conditional likelihood respectively:

$$\mathcal{F}(\mathcal{G};\mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) = \sum_n \log P(\boldsymbol{x}_n|\mathcal{G}) \tag{5.7}$$

and

$$\mathcal{F}(\mathcal{G};\mathcal{D}) = \log P(\mathcal{W}|\mathcal{X},\mathcal{G}) = \sum_n \log P(w_n|\boldsymbol{x}_n,\mathcal{G}) \tag{5.8}$$

In the general criterion (5.6), the criterion function $\mathcal{F}(\mathcal{G};\mathcal{D})$ is allowed to have various forms (real functions). One alternative is the hinge loss function, and the resulting criterion is a large margin training criterion which will be discussed in the following parts.

Let the criterion function $\mathcal{F}(\mathcal{G};\mathcal{D})$ be a hinge loss function:

$$\mathcal{F}(\mathcal{G};\mathcal{D}) = -\sum_n \left[\max_{w \neq w_n} \left\{ \mathcal{L}(w,w_n) - \log\left(\frac{P(w_n|\boldsymbol{x}_n,\mathcal{G})}{P(w|\boldsymbol{x}_n,\mathcal{G})}\right)\right\}\right]_+ \tag{5.9}$$

where $\mathcal{L}(w, w_n)$ is the loss function, which measures how different the labels w and w_n are, and $\left[\,\cdot\,\right]_+$ is the hinge loss, which satisfies:

$$\left[f(x)\right]_+ = \begin{cases} 0 & \text{when } f(x) < 0 \\ f(x) & \text{when } f(x) \geq 0 \end{cases} \tag{5.10}$$

When the criterion function is the hinge loss function (5.9), the general criterion described in (5.6) becomes a large margin training criterion:

$$\begin{aligned} \underset{q(\mathcal{G})}{\arg\min} \Bigg\{ \mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G})\big) + \int q(\mathcal{G}) \sum_n \Bigg[\max_{w \neq w_n} \Big\{ \mathcal{L}(w, w_n) - \\ \log\left(\frac{P(w_n|\boldsymbol{x}_n, \mathcal{G})}{P(w|\boldsymbol{x}_n, \mathcal{G})}\right) \Big\} \Bigg]_+ d\mathcal{G} \Bigg\} \\ \text{s.t.} \quad q(\mathcal{G}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{5.11}$$

This large margin training criterion is related to the maximum entropy discrimination (MED) criterion [90, 93], which is discussed in appendix B.1.1. The main difference is that, in the MED criterion, the integral over the model parameters $\mathcal{G}$ is inside the maximisation. Since the maximum of the integral of a function is less than or equal to the integral of the maximum of that function, the large margin training criterion in (5.11) is an upper bound of the MED criterion. In criterion (5.11), the maximisation is inside the integral, hence the maximums need to be found for all possible $\mathcal{G}$. This minimisation criterion is usually computationally intractable. Thus, approximations to this criterion need to be applied.

One approximate method is to assume the distribution $q(\mathcal{G})$ is a Dirac delta function, namely $q(\mathcal{G}) \approx \delta(\mathcal{G}, \hat{\mathcal{G}})$ with parameters $\hat{\mathcal{G}}$. Substituting this delta function into criterion (5.11), yields:

$$\begin{aligned} \underset{q(\mathcal{G})}{\arg\min} \Bigg\{ \int \delta(\mathcal{G}, \hat{\mathcal{G}}) \log \delta(\mathcal{G}, \hat{\mathcal{G}}) d\mathcal{G} - \log p(\hat{\mathcal{G}}) + \sum_n \Bigg[\max_{w \neq w_n} \Big\{ \mathcal{L}(w, w_n) - \\ \log\left(\frac{P(w_n|\boldsymbol{x}_n, \hat{\mathcal{G}})}{P(w|\boldsymbol{x}_n, \hat{\mathcal{G}})}\right) \Big\} \Bigg]_+ \Bigg\} \end{aligned} \tag{5.12}$$

The first term is the negative of the delta function's entropy, which is an infinite, but constant, value. Then the criterion to be minimised becomes:

$$-\log p(\hat{\mathcal{G}}) + \sum_n \left[\max_{w \neq w_n} \left\{ \mathcal{L}(w, w_n) - \log\left(\frac{P(w_n|\boldsymbol{x}_n, \hat{\mathcal{G}})}{P(w|\boldsymbol{x}_n, \hat{\mathcal{G}})}\right) \right\} \right]_+ \tag{5.13}$$

This is the large margin training criterion for the discriminative models described in [209, 210]. When $P(w|\boldsymbol{x}, \mathcal{G})$ is a log-linear model, the denominator terms of the log-linear model

can be cancelled out in criterion (5.13). This criterion becomes the training criterion of the structured SVM discussed in section 3.2.5, and can be efficiently solved with the cutting-plane algorithm [96].

5.2.1 Solutions to the General Criterion

It is possible to consider a general solution the optimisation criterion (5.6) in the previous section. The criterion function $\mathcal{F}(\mathcal{G};\mathcal{D})$ can be written as $\mathcal{F}(\mathcal{G};\mathcal{D}) = \log\big(\exp\big(\mathcal{F}(\mathcal{G};\mathcal{D})\big)\big)$, hence the general criterion described in (5.6) can be expressed as:

$$\begin{aligned} &\arg\min_{q(\mathcal{G})} \Big\{ \underbrace{\mathrm{KL}\big(q(\mathcal{G})||p(\mathcal{G})\big)}_{\text{prior}} - \underbrace{\int q(\mathcal{G}) \log\big(\exp\big(\mathcal{F}(\mathcal{G};\mathcal{D})\big)\big) d\mathcal{G}}_{\text{evidence}} \Big\} \\ &\text{s.t.} \quad q(\mathcal{G}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{5.14}$$

Analogous with the solutions to the criteria (5.2) and (5.4) for standard Bayesian inference, the solution that minimises the general criterion described in (5.14) can be written as the product of the prior and the exponential of the criterion function:

$$\hat{q}(\mathcal{G}) \propto p(\mathcal{G}) \exp\big(\mathcal{F}(\mathcal{G};\mathcal{D})\big) \tag{5.15}$$

In order to ensure the right hand side of (5.15) is normalisable, the integral over $\exp\big(\mathcal{F}(\mathcal{G};\mathcal{D})\big)$ should be finite. To distinguish from the posterior distribution $p(\mathcal{G}|\mathcal{D})$ obtained from Bayes' rule, the distribution $\hat{q}(\mathcal{G})$ that minimises the general criterion (5.14) is called the *optimal distribution* in this thesis.

In the rest of this section, discriminative models will be used, but similar conclusions can be drawn for generative models. When making predictions, given the optimal distribution $\hat{q}(\mathcal{G})$, the *class posterior distribution* can be obtained by marginalising out all the model parameters [17, 92, 105]:

$$P(w|\boldsymbol{x},\mathcal{D}) = \int P(w|\boldsymbol{x},\mathcal{G})\hat{q}(\mathcal{G})d\mathcal{G} \tag{5.16}$$

However when calculating this class posterior distribution, normally, neither the optimal distribution $\hat{q}(\mathcal{G})$ described in (5.15) nor the integral in (5.16) is tractable. Thus, approximate methods need to be applied. Two approaches will be briefly discussed in the rest of this section.

Monte Carlo Methods Monte Carlo methods are sampling based schemes, in which the samples $\{\mathcal{G}^{(k)}\}_{k=1}^{K}$ are drawn from the optimal distribution $\hat{q}(\mathcal{G})$. The integral in the class posterior distribution described in (5.16) then can be approximated by summing over these samples:

$$P(w|\boldsymbol{x},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K} P(w|\boldsymbol{x},\mathcal{G}^{(k)}) \tag{5.17}$$

Gibbs sampling is one form of the Markov chain Monte Carlo methods [6], which are a class of algorithms for sampling from a probability distribution based on constructing a Markov chain whose equilibrium distribution is the desired distribution. Rather than sampling from the joint distribution $\hat{q}(\mathcal{G}) = \hat{q}(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_M)$ directly (where the model parameters $\mathcal{G}$ can be decomposed as $\mathcal{G} = \{\boldsymbol{g}_1,\ldots,\boldsymbol{g}_M\}$), in Gibbs sampling samples are iteratively drawn from the conditional distribution of each parameter $\boldsymbol{g}_m$ in turn. Following a sufficient burn-in period (say T steps), the chain converges to the stationary distribution regardless of where it begins. This is called the equilibrium distribution. Then, the samples[1] are from the joint distribution $\hat{q}(\mathcal{G})$, but not independent.

In addition to implementing the Monte Carlo method directly, in [95] the joint distribution $q(\mathcal{G})$ is factorised, and the Monte Carlo approach is applied to one of the factorised distributions. Compared with the variational methods having full factorisation, a weaker assumption is made:

$$q(\mathcal{G}) \approx q(\boldsymbol{g_m})q(\mathcal{G}_{-\boldsymbol{g}_m}) \tag{5.18}$$

where $\boldsymbol{g}_m$ is one parameter in set $\mathcal{G} = \{\boldsymbol{g}_1,\ldots,\boldsymbol{g}_M\}$, and $\mathcal{G}_{-\boldsymbol{g}_m}$ are all the parameters except $\boldsymbol{g}_m$ in $\mathcal{G}$. By using this factorisation, inference can be simplified, especially when it is impractical to sample from the joint distribution $q(\mathcal{G})$ directly. The optimisation procedure then can be described as alternatively performing the following two steps:

- Given $\hat{q}(\boldsymbol{g}_m)$, estimate $\hat{q}(\mathcal{G}_{-\boldsymbol{g}_m})$ by minimising the general criterion in (5.6).
- Given $\hat{q}(\mathcal{G}_{-\boldsymbol{g}_m})$, estimate $\hat{q}(\boldsymbol{g}_m)$ by minimising the general criterion in (5.6).

[1] The consecutive samples are correlated with each other, and independent samples are desired. Thus, these samples are thinned by only storing every lth value after the burn-in period.

Often the distribution of a single parameter $\hat{q}(\boldsymbol{g}_m)$ has a simple form, whereas the distribution $\hat{q}(\mathcal{G}_{-\boldsymbol{g}_m})$ may be complicated and have no closed form. Monte Carlo approaches can be implemented [95], where the distribution $\hat{q}(\mathcal{G}_{-\boldsymbol{g}_m})$ is approximated by samples. Then the integral in the class posterior distribution described in (5.16) can be calculated. In section 5.4, an application of this method to the infinite structured discriminative model will be discussed in detail.

Variational Inference As an alternative to Monte Carlo approaches, variational inference can be applied to approximate the optimal distribution $\hat{q}(\mathcal{G})$. The mean field variational inference [19] is commonly used, in which parameters $\mathcal{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M\}$ are assumed to be independent from each other. The optimal distribution $\hat{q}(\mathcal{G})$ is approximated by a fully factorised variational distribution $q_v(\mathcal{G})$:

$$\hat{q}(\mathcal{G}) \approx q_v(\mathcal{G}) = q_v(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M) = \prod_{m=1}^{M} q_v(\boldsymbol{g}_m) \tag{5.19}$$

where the form of each variational distribution $q_v(\boldsymbol{g}_m)$ is determined by model parameter $\boldsymbol{v}_m$. In inference, each parameter $\boldsymbol{v}_m$ is updated in turn iteratively by performing coordinate descent of the general criterion described in (5.6). After converging, the optimised variational distribution can be used in computing the class posterior distribution described in (5.16). Since the variational distribution is fully factorised and normally has a simple form (say the exponential family), the integral becomes tractable to compute. An example of variational inference for the infinite support vector machine is given in Appendix B.

5.3 Infinite Structured Discriminative Models

The framework for infinite mixtures of experts was discussed in section 4.3.5. In the following sections, the applications of infinite mixtures of experts to speech recognition will be discussed in detail. In the previous sections, the input and corresponding label were denoted as $\boldsymbol{x}$ and w respectively. Since speech utterances are data sequences and class labels are sentences, the notation for the input utterance (observation sequence) and corresponding class label (sentence) are changed to $\boldsymbol{O}$ and W respectively. When label structure[1] is

[1] The sentences are structured labels, which can be broken down into atomic units, e.g. words or phones.

introduced, the infinite mixture of experts described in equation (4.47) becomes an *infinite structured discriminative model*:

$$\begin{aligned} P(W|\boldsymbol{O},\mathcal{G}) &= P(W|\boldsymbol{O},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{H}) \\ &= \sum_{z} P(W|\boldsymbol{O},\boldsymbol{H},z)P(z|\boldsymbol{O},\boldsymbol{\pi},\boldsymbol{\Theta}) \quad z \in \{1,2,\cdots,\infty\} \end{aligned} \tag{5.20}$$

where $\mathcal{G} = \{\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{H}\}$ are the parameters of the whole model. z is the indicator variable, that denotes which expert the input variable $\boldsymbol{O}$ is associated with. $P(z|\boldsymbol{O},\boldsymbol{\pi},\boldsymbol{\Theta})$ is known as the *gating network*, which is a function of the input variable $\boldsymbol{O}$ with parameters $\{\boldsymbol{\pi},\boldsymbol{\Theta}\}$. If the gating network is based on a Gaussian mixture model (GMM) with infinite number of components, $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are mixture weights, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ are the parameters of all the components. $P(z|\boldsymbol{O},\boldsymbol{\pi},\boldsymbol{\Theta})$ is the component posterior of the infinite GMM:

$$P(z|\boldsymbol{O},\boldsymbol{\pi},\boldsymbol{\Theta}) = \frac{\pi_z \mathcal{N}\big(\varphi(\boldsymbol{O});\boldsymbol{\theta}_z\big)}{\sum_z \pi_z \mathcal{N}\big(\varphi(\boldsymbol{O});\boldsymbol{\theta}_z\big)}, \quad z \in \{1,2,\ldots,\infty\} \tag{5.21}$$

where $\mathcal{N}\big(\varphi(\boldsymbol{O});\boldsymbol{\theta}_z\big)$ is the zth Gaussian component with parameters $\boldsymbol{\theta}_z$, π_z is the mixture weight corresponding to the zth Gaussian component, and $\varphi(\boldsymbol{O})$ is a feature function, which maps the input $\boldsymbol{O}$ with various length to a feature with fixed dimension, e.g. the log-likelihood feature function described in section 3.5.2.1 can be used.

In the infinite structured discriminative model described in (5.20), $P(W|\boldsymbol{O},\boldsymbol{H},z)$ is a structured discriminative model[1], which is discussed in section 3.2. $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of all the experts. Since the class label of the utterance W is a sentence, the possible number of classes for an utterance can be exponentially large in the vocabulary size. In order to solve this problem, structure is introduced by breaking the sentence label into sub-sentence units, e.g. words or phones with associated segmentation of the sentence. Given one possible segmentation (or alignment) $\boldsymbol{\rho}$ which segments the sentence into sub-sentence units, the input utterance and sentence can be decomposed into $\boldsymbol{O} = \{\boldsymbol{O}_{(1)},\ldots,\boldsymbol{O}_{(|\boldsymbol{\rho}|)}\}$ and $W = \{w_1,\ldots,w_{|\boldsymbol{\rho}|}\}$, where $|\boldsymbol{\rho}|$ is the number of segments. As discussed in section 3.2, the structured discriminative model $P(W|\boldsymbol{O},\boldsymbol{H},z)$ in (5.20) can be described as:

$$P(W|\boldsymbol{O},\boldsymbol{H},z) = P(W|\boldsymbol{O},\boldsymbol{\eta}_z) = \frac{1}{\mathcal{Z}(\boldsymbol{\eta}_z,\boldsymbol{O})} \sum_{\boldsymbol{\rho}\in\boldsymbol{P}_W} \exp\Big(\boldsymbol{\eta}_z^{\mathsf{T}}\Phi(\boldsymbol{O},W,\boldsymbol{\rho})\Big) \tag{5.22}$$

[1] In structured discriminative models, the structure of the class label is considered, and the parameters for any classes (sentences) can be constructed from a common set of basic units [209].

where $\mathcal{Z}(\boldsymbol{\eta}_z, \boldsymbol{O})$ is a normalisation term:

$$\mathcal{Z}(\boldsymbol{\eta}_z, \boldsymbol{O}) = \sum_{W \in \mathcal{W}} \sum_{\boldsymbol{\rho} \in \boldsymbol{P}_W} \exp\left(\boldsymbol{\eta}_z^{\mathsf{T}} \Phi(\boldsymbol{O}, W, \boldsymbol{\rho})\right) \tag{5.23}$$

where the set $\mathcal{W}$ consists of all possible hypotheses for the input $\boldsymbol{O}$, and the set $\boldsymbol{P}_W$ consists of all possible segmentations corresponding to the hypothesis W. $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$ is the *joint feature space*, which characterises the dependence between the input $\boldsymbol{O}$ and hypothesis W, and maps the input $\boldsymbol{O}$ with variable length to a fixed dimension [209, 210]. Various forms of the joint features were discussed in detail in section 3.5.

The structured discriminative model described in equation (5.22) is the conditional augmented (CAug) model or segmental CRF [106, 109, 215] discussed in section 3.2.3. In this model, the summation over all possible segmentations results in inefficiency in training. Similar to Viterbi decoding, where the likelihood is approximated by only considering the most likely state sequence, here the most likely segmentation $\boldsymbol{\rho}_{\boldsymbol{\lambda}}$ from the HMM is used instead of summing over all possible segmentations [147, 199]. Then the structured discriminative model described in (5.22) can be approximated as a *log-linear model*:

$$P(W|\boldsymbol{O}, \boldsymbol{H}, z) = P(W|\boldsymbol{O}, \boldsymbol{\eta}_z) \approx \frac{1}{\mathcal{Z}(\boldsymbol{\eta}_z, \boldsymbol{O})} \exp\left(\boldsymbol{\eta}_z^{\mathsf{T}} \Phi(\boldsymbol{O}, W, \boldsymbol{\rho}_{\boldsymbol{\lambda}})\right) \tag{5.24}$$

When the experts are log-linear models, the infinite structured discriminative model described in (5.20) is an *infinite log-linear model*. Given the generative model (HMM) with parameters $\boldsymbol{\lambda}$, the most likely segmentation $\boldsymbol{\rho}_{\boldsymbol{\lambda}}$ can be obtained by maximising [54]:

$$\boldsymbol{\rho}_{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\rho}} P(\boldsymbol{\rho}) p(\boldsymbol{O}|\boldsymbol{\lambda}, \boldsymbol{\rho}) \tag{5.25}$$

where $p(\boldsymbol{O}|\boldsymbol{\lambda}, \boldsymbol{\rho})$ is the likelihood given by the HMM. In this work, the probabilities of choosing different segments are supposed to be equal, namely $P(\boldsymbol{\rho})$ is a uniform distribution. Alternatively, the optimal segmentation can also be obtained from discriminative models [210]. The optimal segmentation is not considered in this thesis.

5.3.1 *Bayesian Inference with Gibbs Sampling*

The previous section discussed the infinite structured discriminative model, which is an infinite mixture of experts with structured discriminative model experts. The log-linear model, which is a simplified form of the CAug model, was also studied. This section will

discuss Bayesian inference of the infinite structured discriminative model, and the infinite log-linear model[1] will be given as an example.

5.3.1.1 Classification

In order to motivate inference, classification will be discussed first. In classification, given an input utterance $\boldsymbol{O}$, the corresponding class label (sentence) W needs to be predicted. Consider a training set consisting of utterance and reference pairs $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, where $\boldsymbol{O}_n$ is the nth training utterance (observation sequence) and W_n is the corresponding class label. As discussed in section 4.2, in Bayesian approaches the class posterior distribution of W can be calculated by marginalising out over all the model parameters. Thus the class posterior distribution for the infinite structured discriminative model can be obtained:

$$P(W|\boldsymbol{O}, \mathcal{D}) = \int P(W|\boldsymbol{O}, \mathcal{G}) p(\mathcal{G}|\mathcal{D}) d\mathcal{G} \tag{5.26}$$

where $\mathcal{G} = \{\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}\} = \{\pi_m, \boldsymbol{\theta}_m, \boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of the whole model, and the posterior distribution $p(\mathcal{G}|\mathcal{D})$ can be obtained according to Bayes' rule:

$$p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G}) p(\mathcal{D}|\mathcal{G}) \tag{5.27}$$

where $p(\mathcal{G})$ is the prior distribution of the model parameters, and in this work these model parameters are assumed to be independent from each other in the prior distribution. Often, the posterior distribution (5.27) does not have a closed form, and the integral in the class posterior distribution (5.26) is intractable. Common approaches to address this problem are to use Monte Carlo methods or variational inference. In variational inference, the truncation is made in inference [19, 213], where the number of components is truncated to M. And in the mean field variational inference which is most commonly used, the fully-factorized variational distributions are used which break the dependencies between the parameters. Alternatively, Monte Carlo approaches are sampling-based methods, where the integral, such as in (5.26), is approximated by the sum over samples, and any desired accuracy can be achieved with enough samples. In this work, only Monte Carlo approaches are studied.

[1] Again, the infinite log-linear model is an infinite structured discriminative model having log-linear models as experts.

Given the definition of the infinite structured discriminative model (5.20), and let the integral in the class posterior distribution (5.26) be approximated by the sum over K samples, then yields:

$$\begin{aligned} P(W|\boldsymbol{O},\mathcal{D}) &\approx \frac{1}{K}\sum_{k=1}^{K} P(W|\boldsymbol{O},\mathcal{G}^{(k)}) \\ &\approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} P(W|\boldsymbol{O},\boldsymbol{H}^{(k)},z)P(z|\boldsymbol{O},\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)}) \end{aligned} \tag{5.28}$$

where the samples $\mathcal{G}^{(k)} = \{\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)},\boldsymbol{H}^{(k)}\}_{k=1}^{K}$ are drawn from the posterior distribution of the model parameters $p(\mathcal{G}|\mathcal{D})$. Since $\mathcal{G} = \{\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{H}\}$ are the parameters of the whole infinite structured discriminative model, there are infinite number of parameters in set $\{\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{H}\}$. In training (limited by the finite number of training data), the number of *represented experts* M_k (which are the experts that have associated data) is finite. In classification, only the represented experts are considered (and the reason will be discussed in the next paragraph). Then each draw $\{\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)},\boldsymbol{H}^{(k)}\} \approx \{\pi_m^{(k)},\boldsymbol{\theta}_m^{(k)},\boldsymbol{\eta}_m^{(k)}\}_{m=1}^{M_k}$ [1] defines a mixture of experts as described in section 4.3.5.1, and classification is averaged over K mixture of experts. In the rest of this section, the reason for only considering the represented experts will be discussed.

In the class posterior distribution (5.28), M_k is the number of the represented experts (which are the experts that have associated data) for the kth draw. This means the unrepresented experts (which are the experts that have no associated data) are ignored in classification. Since there is no associated data for the unrepresented experts, the parameters of the unrepresented experts are sampled from the prior distribution, and all unrepresented experts can be treated as a single expert. Thus, in the following discussion, we can consider the unrepresented experts as a single expert. For the kth draw, the conditional probability $P_{\text{unrep}}(W|\boldsymbol{O},\mathcal{G}^{(k)})$ contributed from the unrepresented experts can be described as:

$$\begin{aligned} P_{\text{unrep}}(W|\boldsymbol{O},\mathcal{G}^{(k)}) &= P(W|\boldsymbol{O},\boldsymbol{H}^{(k)},z)P(z|\boldsymbol{O},\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)}) \\ &= P(W|\boldsymbol{O},\boldsymbol{\eta}_z^{(k)})P(z|\boldsymbol{O},\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)}), \qquad z = M_k+1 \end{aligned} \tag{5.29}$$

[1] The mth mixture weight $\pi_m^{(k)}$ is proportional to the number of data associated with that expert $N_m^{(k)}$ which can be obtained in training, and the total weight satisfies $\sum_m \pi_m^{(k)} = 1$.

Let $\pi^{(k)}_{M_k+1}$ [1] denote the mixture weight corresponding to the unrepresented experts. Empirically, the mixture weight $\pi^{(k)}_{M_k+1}$ corresponding to the unrepresented experts is small. This leads to a small component posterior probability $P(z|\boldsymbol{O}, \boldsymbol{\pi}^{(k)}, \boldsymbol{\Theta}^{(k)})$ defined in (5.21). Moreover, if the unrepresented experts are considered in classification, in the conditional distribution $P(W|\boldsymbol{O}, \boldsymbol{\eta}^{(k)}_z)$, $\boldsymbol{\eta}^{(k)}_z$ is drawn the prior distribution $p(\boldsymbol{\eta})$, and the prior might be chosen for mathematical simplicity rather than based on the prior beliefs or we hold no prior information about the parameter. This leads to inaccuracy when the unrepresented experts are taken into account. Due to the small value of $P(z|\boldsymbol{O}, \boldsymbol{\pi}^{(k)}, \boldsymbol{\Theta}^{(k)})$ and inaccuracy of the conditional probability $P_{\text{unrep}}(W|\boldsymbol{O}, \boldsymbol{\eta}^{(k)}_z)$ described in (5.29), the unrepresented experts are not considered in classification in this study.

5.3.1.2 *Bayesian Inference*

As discussed in the previous section, the joint posterior distribution over the model parameters, $p(\mathcal{G}|\mathcal{D})$, of the infinite structured discriminative model described in (5.27) does not have a closed form. However, Gibbs sampling [6, 190] can be applied to draw samples from this joint posterior distribution. In inference of the infinite structured discriminative model, the representation based on the Chinese restaurant process (CRP) is used, and the graphical model of this representation is illustrated in Figure 4.10, where the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are marginalised out, and the latent variables (the indicator variables corresponding to the training data) $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ are introduced. Thus, the posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D})$ will be inferred in training. Gibbs sampling is applied to draw samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}, \boldsymbol{z}^{(k)}\}$ from this posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D})$. As discussed in the previous section, $\{\boldsymbol{\pi}^{(k)}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}\}$ are the samples of interest in classification. As discussed in [6, 172], the marginal samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}\}$ can be obtained by sampling $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}, \boldsymbol{z}^{(k)}\}$ according to $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D})$ and subsequently ignoring the samples $\boldsymbol{z}^{(k)}$. When the sampled indicators $\boldsymbol{z}^{(k)}$ are given, the number of the represented experts M_k can be determined, which is the number of the unique values in set $\boldsymbol{z}^{(k)}$, namely $M_k = |\boldsymbol{z}^{(k)}|$. Given $\boldsymbol{z}^{(k)}$, the number of data $N^{(k)}_m$ associated with each

[1] When the unrepresented experts are considered, the mixture weight $\pi^{(k)}_{M_k+1}$ corresponding to the unrepresented experts is proportional to α which is the concentration parameter of the Dirichlet process; the weight $\pi^{(k)}_m$ corresponding to a represented expert is proportional to the number of data associated with that expert $N^{(k)}_m$. The total weight satisfies $\sum_m \pi^{(k)}_m = 1$.

Algorithm 1: the sampling process for the infinite structured discriminative model

Initialise: $\{\boldsymbol{\Theta}^{(0)}, \boldsymbol{H}^{(0)}, \boldsymbol{z}^{(0)}\}$
repeat
  for $n \in \{1, \ldots, N\}$ **do**
    sample $z_n^{(k)} \sim P(z_n | \boldsymbol{z}_{-n}^{(k)}, \boldsymbol{\Theta}^{(k-1)}, \boldsymbol{H}^{(k-1)}, \mathcal{D})$
  end
  The number of represented experts is: $M_k = |\boldsymbol{z}^{(k)}|$.
  for $m \in \{1, \ldots, M_k\}$ **do**
    sample $\boldsymbol{\theta}_m^{(k)} \sim p(\boldsymbol{\theta}_m | \boldsymbol{z}^{(k)}, \mathcal{D})$
    sample $\boldsymbol{\eta}_m^{(k)} \sim p(\boldsymbol{\eta}_m | \boldsymbol{z}^{(k)}, \mathcal{D})$
  end
  for *the unrepresented experts* **do**
    sample $\boldsymbol{\theta}_{M_k+1}^{(k)} \sim p(\boldsymbol{\theta})$
    $\boldsymbol{\eta}_{M_k+1}^{(k)} = \arg\max_{\boldsymbol{\eta}} p(\boldsymbol{\eta})$[1]
  end
until *converge*;

represented expert also can be determined: $N_m^{(k)} = \sum_{n=1}^{N} \delta(z_n^{(k)}, m)$. As discussed in section 5.3.1.1, only the represented experts are considered in classification, the mixture weights $\boldsymbol{\pi}^{(k)} = \{\pi_m^{(k)}\}_{m=1}^{M_k}$ [2] used in classification then can be determined, where each weight $\pi_m^{(k)}$ is proportional to the number of associated data $N_m^{(k)}$, and the total weight satisfies $\sum_m \pi_m^{(k)} = 1$, namely $\pi_m^{(k)} = \frac{N_m^{(k)}}{N}$ where N is the total number of training data.

In Gibbs sampling, samples are iteratively drawn from the conditional posterior distribution of each parameter in turn. The sampling process for the infinite structured discriminative model can be summarised in Algorithm 1. The samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}, \boldsymbol{z}^{(k)}\}$ can be obtained from this iterative process. By discarding the initial set of samples (in burn-in period) to avoid starting biases, K thinned samples (obtained by only choosing one sample from every several consecutive samples) can be used to approximate the integral in the class posterior distribution described in (5.26). In the following sections, the conditional posterior distribution of each parameter will be discussed in detail.

[2] Since only the represented experts are considered in classification, the mixture weights $\boldsymbol{\pi}^{(k)}$ has M_k elements. Training is based on the CRP, where $\boldsymbol{\pi}$ are marginalised out, and infinite number of experts are considered. Thus, there is no restriction on training.

5.3.1.3 The Conditional Posterior Distribution of z_n

The indicator variable z_n denotes which expert the nth observation sequence $\boldsymbol{O}_n$ is associated with. Given the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$ and the samples of all the other model parameters, the conditional posterior distribution of z_n can be described as follows:

$$P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \boldsymbol{\Theta}^{(k-1)}, \boldsymbol{H}^{(k-1)}, \mathcal{D}) \propto P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha) p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big) P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m^{(k-1)}) \tag{5.30}$$

where the set $\boldsymbol{z}_{-n}^{(k)}$ denotes all the indicators except $z_n^{(k)}$, namely

$$\boldsymbol{z}_{-n}^{(k)} = \{z_1^{(k)}, \ldots, z_{n-1}^{(k)}, z_{n+1}^{(k-1)}, \ldots, z_N^{(k-1)}\}.$$

$\boldsymbol{\Theta}^{(k-1)}$ are the sampled parameters of the components in the gating network, and $\boldsymbol{H}^{(k-1)}$ are the sampled parameters of the experts. The first term $P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha)$ is given by the *Chinese Restaurant Process* (CRP)[1] with concentration parameter α described in section 4.3.3:

$$P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha) = \begin{cases} \dfrac{N_{m,-n}^{(k)}}{N - 1 + \alpha}, & \text{when } m \text{ denotes an existing expert} \\ \dfrac{\alpha}{N - 1 + \alpha}, & \text{when } m \text{ denotes a new expert} \end{cases} \tag{5.31}$$

where N is the total number of the training data, and $N_{m,-n}^{(k)}$ is the number of training data associated with the mth expert excluding the nth instance. In (5.30) the second term $p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big)$ is the component likelihood, which is given by the Gaussian distribution. $\varphi(\cdot)$ is a feature function which transforms the input with various length to a vector with fixed dimension. The last term $P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m^{(k-1)})$ is the conditional likelihood given by the structured discriminative model, for example the log-linear model described in (5.24).

When z_n indicates an existing expert (a represented expert), it is straightforward to calculate the conditional posterior probability of z_n through (5.30). When z_n denotes a new expert (an unrepresented expert), following the method introduced by Neal [131], when

[1] In order to make the newly generated expert have good generalisation, $\boldsymbol{\eta}$ is set to be the mode of its prior distribution. Here, the mode of the prior distribution is the optimised parameter of the discriminative model trained with the whole training set.

[1] Given the exchangeability of the CRP, each z_n can be considered as the last customer in the CRP metaphor.

calculating the likelihood $p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big)$, the parameter $\boldsymbol{\theta}_m^{(k-1)}$ is sampled from its prior distribution as an auxiliary parameter [131]. Thus the likelihood can be easily obtained. In order to ensure that the newly generated expert has good generalisation, when calculating the last term $P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m^{(k-1)})$, the parameter of the expert $\boldsymbol{\eta}_m^{(k-1)}$ is set to be the mode of the prior distribution. In this work, the prior distribution is a Gaussian distribution, and the mean of this distribution is the optimised parameter of the structured discriminative model trained with the whole training set.

5.3.1.4 The Conditional Posterior Distribution of $\boldsymbol{\theta}_m$

The gating network is based on a Dirichlet process mixture model, i.e. an infinite GMM in this work. Given the sampled indicator variables $\boldsymbol{z}^{(k)}$, the parameters of different Gaussian components are conditionally independent. Here the conditional posterior distribution of the mth Gaussian component parameters $\boldsymbol{\theta}_m$ can be described as follows:

$$p(\boldsymbol{\theta}_m|\boldsymbol{z}^{(k)}, \mathcal{D}) \propto p(\boldsymbol{\theta}_m) \prod_{\forall z_n^{(k)}=m} p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m\big) \tag{5.32}$$

where $\forall z_n^{(k)} = m$ indicates all instances $n \in \{1, \ldots, N\}$ that satisfy $z_n^{(k)} = m$. $p(\boldsymbol{\theta}_m)$ is the prior distribution of $\boldsymbol{\theta}_m$, and $p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m\big)$ is the likelihood corresponding to the mth Gaussian component. Given the sampled indicators $\boldsymbol{z}^{(k)}$, each parameter set of the Gaussian component $\boldsymbol{\theta}_m^{(k)}$ can be sampled independently according to the conditional posterior distribution in (5.32). This conditional posterior distribution (5.32) is the same as the conditional posterior distribution of the component parameters in the infinite GMM [149], and the detailed process of sampling the component parameters and their hyper parameters is described in [77, 131, 149].

5.3.1.5 The Conditional Posterior Distribution of $\boldsymbol{\eta}_m$

In this section, the experts are assumed to be log-linear models described in (5.24). Given the sampled indicator variables $\boldsymbol{z}^{(k)}$, the parameters of different experts are conditionally independent. Then the conditional posterior distribution of the mth expert parameter $\boldsymbol{\eta}_m$ can be described as follows:

$$p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D}) \propto p(\boldsymbol{\eta}_m) \prod_{\forall z_n^{(k)}=m} P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m) \tag{5.33}$$

Algorithm 2: the Metropolis algorithm for sampling the parameter of the expert

Initialise: $\boldsymbol{\eta}_0, t = 0$

repeat

1. Sample a candidate from the *proposal distribution* $\boldsymbol{\eta}^* \sim p(\boldsymbol{\eta}|\boldsymbol{\eta}_t)$, which is a symmetric distribution satisfying $p(\boldsymbol{\eta}|\boldsymbol{\eta}_t) = p(\boldsymbol{\eta}_t|\boldsymbol{\eta})$.

2. Calculate the ratio:

$$r = \frac{p(\boldsymbol{\eta}^*|\boldsymbol{z}^{(k)}, \mathcal{D})}{p(\boldsymbol{\eta}_t|\boldsymbol{z}^{(k)}, \mathcal{D})} = \frac{p(\boldsymbol{\eta}^*) \prod_{\forall z_n^{(k)}=m} P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}^*)}{p(\boldsymbol{\eta}_t) \prod_{\forall z_n^{(k)}=m} P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_t)} \tag{5.34}$$

3. If $r \geq 1$, accept the candidate. If $r < 1$, with probability r accept the candidate, else reject it and go to step 1.

4. Let $t = t + 1$, and $\boldsymbol{\eta}_t = \boldsymbol{\eta}^*$.

until *converge*;

where $\forall z_n^{(k)} = m$ indicates all instances $n \in \{1, \ldots, N\}$ that satisfy $z_n^{(k)} = m$. $p(\boldsymbol{\eta}_m)$ is the prior distribution of the expert parameter, which is a Gaussian distribution $p(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\eta}_m; \boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_\eta)$. In this work, the mean $\boldsymbol{\mu}_\eta$ is the optimised parameter of a structured discriminative model trained with the whole training set, and the covariance is a scaled identity matrix $\boldsymbol{\Sigma}_\eta = C\boldsymbol{I}$. $P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m)$ is the conditional likelihood of a structured discriminative model, e.g. the log-linear model described in (5.24). Given the sampled indicators $\boldsymbol{z}^{(k)}$, the parameter of each expert $\boldsymbol{\eta}_m^{(k)}$ can be sampled independently according to the conditional posterior distribution described in (5.33). The Metropolis algorithm [125, 126] can be applied to sample from this conditional posterior distribution, and the sampling process is detailed in Algorithm 2.

As described in Algorithm 2, in step 1 the candidate $\boldsymbol{\eta}^*$ for the next sample is drawn from the *proposal distribution* (or *candidate-generating distribution*) $p(\boldsymbol{\eta}|\boldsymbol{\eta}_t)$, which is a symmetric distribution satisfying $p(\boldsymbol{\eta}|\boldsymbol{\eta}_t) = p(\boldsymbol{\eta}_t|\boldsymbol{\eta})$, e.g. a Gaussian distribution centred at $\boldsymbol{\eta}_t$: $\mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\eta}_t, \Sigma)$. In step 2, when calculating the ratio r described in (5.34), the denominator term of the log-linear model $P(W|\boldsymbol{O}, \boldsymbol{\eta})$ can be calculated efficiently by applying the forward algorithm on the denominator lattice [196].

By repeating the sampling process in Algorithm 2, a Markov chain $\{\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \ldots\}$ can be built. Following the burn-in period (e.g. T steps), the chain approaches its stationary distribution. Then $\{\boldsymbol{\eta}_{T+1}, \boldsymbol{\eta}_{T+2}, \ldots\}$ are samples from distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$.

Each time the ratio r is calculated, N_m (the amount of data associated with the mth expert) calls of the forward algorithm are required to calculate the denominator term of $P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}^*)$. However, the value of the product of the posterior probabilities $\prod_{\forall z_n^{(k)}=m} P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}^*)$ can be cached for the next calculation of r, here $\prod_{\forall z_n^{(k)}=m} P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_t)$ does not need to be calculated, given that the cached value can be used. However, generating a valid sample by using the Metropolis algorithm is still computationally expensive. Since the Metropolis algorithm is a Markov chain Monte Carlo (MCMC) approach, this chain might be slow to converge and the iterative calculation of the ratio r described in (5.34) is computationally expensive. This motivates an approximate approach to be used, and this approach will be discussed in the following section.

5.3.2 *MAP Estimation for Each Expert*

In section 5.3.1, a MCMC approach (i.e. Gibbs sampling detailed in Algorithm 1) is introduced to draw samples from the posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D})$. When sampling from the conditional posterior distribution of each expert's parameter $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$, the Metropolis algorithm described in Algorithm 2 is applied. In this algorithm, each instance of calculating the ratio r described in (5.34) requires the forward algorithm to be implemented. This leads to inefficiency in training. In order to make training more efficient, an approximate method will be discussed in this section.

Rather than drawing samples from the conditional posterior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$ described in (5.33), the *maximum a posteriori* (MAP) estimate can be used to approximate the samples from this posterior distribution. The MAP estimate is the mode of the conditional posterior distribution of $\boldsymbol{\eta}_m$ described in (5.33):

$$\begin{aligned}\arg\max_{\boldsymbol{\eta}_m} \mathcal{F}_{\text{MAP}}(\boldsymbol{\eta}_m) &= \arg\max_{\boldsymbol{\eta}_m} \left\{ \log \left(p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D}) \right\} \right. \\ &= \arg\max_{\boldsymbol{\eta}_m} \left\{ \log p(\boldsymbol{\eta}_m) + \sum_{\forall z_n^{(k)}=m} \log P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m) \right\} \end{aligned} \tag{5.35}$$

This is the same as the CML training criterion of the CAug model with a prior [106, 147]. Then the efficient training approaches used by the CAug model can be applied, such as the RPROP algorithm [151]. When the prior is a Gaussian distribution $p(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\eta}_m; \boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_\eta)$

with mean $\boldsymbol{\mu}_\eta$ and a scaled identity covariance matrix $\boldsymbol{\Sigma}_\eta = C\boldsymbol{I}$, the MAP (or CML) estimation described in (5.35) can be further described as:

$$\arg\max_{\boldsymbol{\eta}_m} \mathcal{F}_{\text{MAP}}(\boldsymbol{\eta}_m) = \arg\max_{\boldsymbol{\eta}_m} \Big\{ -\frac{||\boldsymbol{\eta}_m - \boldsymbol{\mu}_\eta||^2}{C} + \sum_{\forall z_n^{(k)}=m} \log P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m) \Big\} \tag{5.36}$$

By using an appropriate prior, improved generalisation can be achieved. In this work, the mean of the prior $\boldsymbol{\mu}_\eta$ is set to be the optimised parameter of the log-linear model (5.24) trained on the whole training set using the CML criterion.

As discussed in this section, in training the infinite log-linear model (iLLM), MAP (or CML) estimation is applied to estimate the parameter of each expert by replacing sampling from the conditional posterior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$ in Algorithm 1.

In this approximate method, the parameter of each expert is estimated according to the MAP (or CML) criterion, and this estimate is used to approximate the sample drawn from the conditional posterior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$. By using this approximation, the samples $\boldsymbol{H}^{(k)}$ drawn from the conditional posterior distributions are replaced with the parameters trained with the MAP criterion described in (5.36). Thus, the samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}\}$ are not drawn from the joint posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{H}|\mathcal{D})$ obtained from Bayes' rule, but from a distribution with more narrow support. The resulting samples might converge to a local maximum of $p(\boldsymbol{\Theta}, \boldsymbol{H}|\mathcal{D})$. This might limit the ability of the model exploring the whole parameter space and mitigating over-fitting.

5.3.3 *Large Margin Training for Each Expert*

In section 5.3.2, the parameters of the experts are obtained by using a MAP estimator. In order to take advantage of the large margin classifier, with good generalisation yielding the state-of-the-art performance, in this section, large margin training is applied to estimate the parameters of each expert by replacing sampling from the conditional posterior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$ in Algorithm 1. This type of system has been called the *infinite structured SVM* in the previous work [197].

In large margin training of the log-linear model described in (5.24), the margin is defined as the log-posterior ratio of the log-linear model between the reference W_n and the most competing hypothesis W. Given the sampled indicators $\boldsymbol{z}^{(k)}$, and introducing a prior

$p(\boldsymbol{\eta}_m)$, the parameter of the mth expert can be estimated by minimising the following large margin training criterion:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\eta}_m) = -\log p(\boldsymbol{\eta}_m) + \sum_{\forall z_n^{(k)}=m,\forall n} \Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W,W_n) - \log\Big(\frac{P(W_n|\boldsymbol{O}_n,\boldsymbol{\eta}_m)}{P(W|\boldsymbol{O}_n,\boldsymbol{\eta}_m)}\Big) \Big\} \Bigg]_+ \tag{5.37}$$

where, for each training instance, the most competing hypothesis and segmentation pair $(W,\boldsymbol{\rho})$ is found over all possible hypotheses and segmentations[1] except the reference with the corresponding segmentation $(W_n,\boldsymbol{\rho}_n)$, and $\boldsymbol{\rho}_n$ is the most likely segmentation obtained by the HMM as described in equation (5.25). $\mathcal{L}(W,W_n)$ is the loss, which measures how different the hypothesis W and the reference W_n are. When the prior $p(\boldsymbol{\eta}_m)$ is a Gaussian distribution $p(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\eta}_m;\boldsymbol{\mu}_\eta,\boldsymbol{\Sigma}_\eta)$ with mean $\boldsymbol{\mu}_m$ and scaled identity covariance matrix $\boldsymbol{\Sigma}_\eta = C\boldsymbol{I}$, substituting the log-linear model (5.24) into the large margin training criterion (5.37), the denominator term of the log-linear model can be cancelled out. Then, the training criterion can be further described as minimising:

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\eta}_m) = \frac{1}{2}||\boldsymbol{\eta}_m - \boldsymbol{\mu}_\eta||^2 + C \sum_{\forall z_n^{(k)}=m,\forall n} \Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W,W_n) + \boldsymbol{\eta}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\boldsymbol{\rho}) \Big\} - \boldsymbol{\eta}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\boldsymbol{\rho}_n) \Bigg]_+ \tag{5.38}$$

This is the training criterion of the structured SVM [180, 181] with the training data associated with the mth expert. In order to train the expert with limited training data and ensure good generalisation, in this work the mean of the Gaussian prior $\boldsymbol{\mu}_\eta$ is set to be the optimised parameter of the structured SVM trained with the whole training set.

As discussed in this section, in Gibbs sampling, sampling for each expert is replaced by large margin training. Thus, this is not a valid Gibbs sampling approach, but an approximation. This type of large margin training for each expert can be viewed as a special example of large margin training for all the experts, which is discussed in detail in Appendix E.

5.3.4 Relationships with the General Criterion

In section 5.3.1, Bayesian inference of the infinite log-linear model was studied. In inference, the framework of infinite mixtures of experts based on the Chinese restaurant pro-

[1] These possible hypotheses and segmentations can be obtained from a denominator lattice [147, 209].

cess (CRP) (illustrated in Figure 4.10) is used, where the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are marginalised out, and the indicator variables $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ corresponding to the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$ are introduced. Thus, the posterior distribution of the model parameters to be inferred is $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D})$, and Gibbs sampling is employed to sample from this distribution. As discussed in section 5.2, the model posterior distribution obtained through Bayes' rule can be interpreted as the application of the general criterion (5.6) with a log-likelihood criterion function:

$$\underset{q(\boldsymbol{\Theta},\boldsymbol{H},\boldsymbol{z})}{\arg\min} \left\{ \mathrm{KL}\big(q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})||p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})\big) - \int \sum_{\boldsymbol{z}} q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) d(\boldsymbol{\Theta}, \boldsymbol{H}) \right\} \tag{5.39}$$

$$\text{s.t.} \quad q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \in \mathcal{P}_{\text{prob}}$$

As discussed in section 4.3.5, the infinite mixture of experts (based on the CRP) is a mixture of discriminative models, but not a discriminative model which models the conditional distribution of the label W given the observations $\boldsymbol{O}$ directly without considering the underlying distribution of the observations [132]. For the infinite log-linear model, the distribution of the observations is modelled by the gating network, which is an infinite Gaussian mixture model. The infinite log-linear model is a combination of generative models (gating network) and discriminative models (experts). Thus, the criterion function criterion function $\mathcal{F}(\Theta, H, \boldsymbol{z}; \mathcal{D})$ can be written in the form of a combination of likelihoods and conditional likelihoods:

$$\begin{aligned} \mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) = \log p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) &= \sum_n \log p(\boldsymbol{O}_n, W_n|\boldsymbol{\Theta}, \boldsymbol{H}, z_n) \\ &= \sum_n \Big(\log p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n) + \log P(W_n|\boldsymbol{O}_n, \boldsymbol{H}, z_n) \Big) \end{aligned} \tag{5.40}$$

As shown shown in (5.15), the solution that minimises criterion (5.39) is the product of the prior and the exponential of the criterion function:

$$\begin{aligned} \hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) &= p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \log \big(\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) \big) \\ &= p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) P(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) = p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}|\mathcal{D}) \end{aligned} \tag{5.41}$$

This is the posterior distribution obtained through Bayes' rule, and it is the distribution to be inferred in standard Bayesian inference discussed in section 5.3.1.

5.4 Large Margin Training of Infinite Log-linear Models

In section 5.3.1, Bayesian inference of the infinite structured discriminative model was studied. Section 5.3.4 discussed a criterion-based interpretation of Bayesian inference for the infinite log-linear model, in which the general criterion has a log-likelihood criterion function. This section will discuss the application of the general criterion to the whole infinite log-linear model, the experts and the gating network.

5.4.1 *The Training Criterion*

As discussed in section 5.3.4, Bayesian inference of the infinite log-linear model can be interpreted as a criterion-based training approach, where the training criterion is the general criterion (5.39) with a log-likelihood criterion function (5.40). This criterion function is the combination of likelihoods (for the gating network) and conditional likelihoods (for the experts) as described in (5.40):

$$\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) = \sum_n \Big(\log p(\boldsymbol{O}_n | \boldsymbol{\Theta}, z_n) + \log P(W_n | \boldsymbol{O}_n, \boldsymbol{H}, z_n) \Big) \tag{5.42}$$

Appendix E discussed (large margin) training of the experts for the infinite log-linear model, where the general criterion with a hinge loss criterion function is applied only to the experts, and this criterion function is described in (E.2):

$$\mathcal{F}(\boldsymbol{H}; \mathcal{D}) = -\sum_n \left[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{H}, \boldsymbol{O}_n) \Big\} \right]_+ \tag{5.43}$$

In this section, a large margin training criterion will be applied to the whole infinite log-linear model. This large margin training criterion is a specification of the general criterion, and has the same form as the general criterion for Bayesian inference described in (5.39):

$$\begin{aligned} &\arg\min_{q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})} \left\{ \mathrm{KL}\big(q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) || p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})\big) - \int \sum_{\boldsymbol{z}} q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) d(\boldsymbol{\Theta}, \boldsymbol{H}) \right\} \\ &\text{s.t.} \quad q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{5.44}$$

but the criterion function has a different form, which is the combination of the likelihood

function (for the gating network) and the hinge loss function (for the experts):

$$\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D}) = \sum_n \Bigg(\log p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n) - \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) \Big\} \Big]_+ \Bigg) \tag{5.45}$$

This criterion function (5.45) is called the *combined criterion function* in this work. The general criterion described in (5.44) with this combined criterion function (5.45) has a similar form to the criterion for *regularised Bayesian inference* [212, 214], and the hinge loss function (the second term of (5.45)) is related to the regularisation term in regularised Bayesian inference.

For the combined criterion function (5.45), $[\cdot]_+$ is the hinge loss function defined in (5.10). In this hinge loss, for each training instance, the best competing hypothesis and segmentation pair $(W, \boldsymbol{\rho})$ is found over all possible hypotheses and segmentations except the reference with the corresponding segmentation $(W_n, \boldsymbol{\rho}_n)$, and $\boldsymbol{\rho}_n$ is the most likely segmentation obtained by the HMM as described in equation (5.25). $\mathcal{L}(W, W_n)$ is the loss between the hypothesis W and the reference W_n. $\mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n)$ is the margin, which determines how well the input label W_n can be correctly separated with the hypothesis W. In the training criterion of the structured SVM [210], the margin is defined as the log-posterior ratio of the log-linear model defined in (5.24). In this margin definition, the normalisation terms of the log-linear models can be cancelled out. Analogously, in this work the margin for the infinite log-linear model is defined as the log ratio of the log-linear models:

$$\mathcal{M}(W_n, W; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = \log \frac{P(W_n|\boldsymbol{O}_n, \boldsymbol{H}, z_n)}{P(W|\boldsymbol{O}_n, \boldsymbol{H}, z_n)} \tag{5.46}$$

where $P(W|\boldsymbol{O}, \boldsymbol{H}, z)$ is the log-linear model defined in (5.24). Other possible definitions of the margin were discussed in appendix E.1.1, where large margin training is applied in training the experts of the infinite log-linear model. In this section, large margin training is applied to the whole infinite log-linear model (an overall training criterion for the gating network and experts).

5.4.2 The Solution to the Criterion

In the previous section, large margin training for the whole infinite log-linear model was discussed. This large margin training criterion is the general criterion (5.44) with the com-

bined criterion function described in (5.45). As discussed in section 5.2.1, the solution that minimises the general criterion is the product of the prior and the exponential of the criterion function. Thus, minimising the general criterion (5.44) with the combined criterion function (5.45), yields:

$$\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \propto p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \exp\left(\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D})\right) \tag{5.47}$$

where $p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})$ is the prior distribution of the model parameters, and in this work these model parameters are assumed to be independent from each other in the prior distribution. Substituting the definition of the combined criterion function (5.45) in, this solution (5.47) can be further written as:

$$\begin{aligned}\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \propto p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \prod_n \Bigg(& p(\boldsymbol{O}_n | \boldsymbol{\Theta}, z_n) \cdot \\ & \exp\Bigg(- \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) \Big\} \Big]_+ \Bigg) \Bigg)\end{aligned} \tag{5.48}$$

This distribution is intractable, hence a Monte Carlo approach is implemented to sample from the un-normalised distribution as illustrated on the right hand side of (5.48). Because of the maximisation in the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})$, it is impractical to draw samples from this distribution directly. As discussed in section 5.2.1, an indirect Monte Carlo method can be applied. In this Monte Carlo approach, the joint posterior distribution is factorised. Then, samples can be drawn from the factorised distributions. In this work, a weak assumption is made [95]:

$$q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \approx q(\boldsymbol{H}) q(\boldsymbol{\Theta}, \boldsymbol{z}) \tag{5.49}$$

Under this assumption, the training process can be summarised in Algorithm 3, and training will be discussed in detailed in the following sections.

5.4.2.1 *The Estimation of* $\hat{q}(\boldsymbol{H})$

Given $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, the optimal distribution $\hat{q}(\boldsymbol{H})$ for the experts' parameters can be estimated by minimising the general criterion (5.44) with the combined criterion function (5.45). Since $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ is given, this general criterion can be simplified as the following large margin

Algorithm 3: Large margin training for the infinite log-linear model

Initialise: $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ and $\hat{q}(\boldsymbol{H})$

repeat

1. Given $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, by minimising the general criterion (5.44) with the combined criterion function (5.45), the optimal distribution $\hat{q}(\boldsymbol{H})$ can be estimated.

2. Given $\hat{q}(\boldsymbol{H})$, the optimal distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ can be obtained by minimising the general criterion (5.44). Then the samples are drawn from the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$.

until *converge*;

training criterion:

$$\begin{aligned}\arg\min_{q(\boldsymbol{H})} \Big\{ \mathrm{KL}\big(q(\boldsymbol{H})||p(\boldsymbol{H})\big) + \int \sum_{\boldsymbol{z}} \hat{q}(\boldsymbol{z}) q(\boldsymbol{H}) \sum_{n=1}^{N} \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \\ \mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) \Big\} \Big]_+ d\boldsymbol{H} \Big\} \\ \text{s.t.} \quad q(\boldsymbol{H}) \in \mathcal{P}_{\text{prob}}\end{aligned} \tag{5.50}$$

where the margin $\mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n)$ is defined in (5.46). Substituting the definition of the log-linear model (5.24) into this margin, the denominator terms of the log-linear models can be cancelled out, then yields:

$$\mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = \boldsymbol{\eta}_{z_n}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) - \boldsymbol{\eta}_{z_n}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \tag{5.51}$$

For the infinite log-linear model, the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ does not have a closed form, hence in training samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}\}$ are drawn from this distribution, which will be discussed in detail in section 5.4.2.2. In the large margin training criterion (5.50), K samples $\{\boldsymbol{z}^{(k)}\}_{k=1}^{K}$ are used to approximate the distribution $\hat{q}(\boldsymbol{z})$, and substituting margin (5.51) in, the large margin training criterion (5.50) then can be approximated as follows:

$$\begin{aligned}\arg\min_{q(\boldsymbol{H})} \Big\{ \mathrm{KL}\big(q(\boldsymbol{H})||p(\boldsymbol{H})\big) + \int \frac{1}{K} \sum_{k=1}^{K} q(\boldsymbol{H}) \sum_{n=1}^{N} \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \\ \boldsymbol{\eta}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) + \boldsymbol{\eta}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \Big\} \Big]_+ d\boldsymbol{H} \Big\} \\ \text{s.t.} \quad q(\boldsymbol{H}) \in \mathcal{P}_{\text{prob}}\end{aligned} \tag{5.52}$$

Since the most competing hypothesis W and corresponding segmentation $\boldsymbol{\rho}$ are found over all possible $\boldsymbol{H}$, this large margin training criterion (5.52) is intractable. Approximations

need to be made. As discussed in section 5.2, the large margin training criterion becomes tractable, when the posterior distribution $q(\boldsymbol{H})$ is assumed to be a Dirac delta function:

$$q(\boldsymbol{H}) = \delta(\boldsymbol{H}, \hat{\boldsymbol{H}}) = \prod_{m=1}^{\infty} \delta(\boldsymbol{\eta}_m, \hat{\boldsymbol{\eta}}_m) \tag{5.53}$$

This delta function has point mass at $\hat{H} = \{\hat{\boldsymbol{\eta}}_m\}_{m=1}^{\infty}$, and they are the parameters of the delta function. Substituting the definition of the KL divergence and this delta function (5.53) in, the large margin training criterion (5.52) can be further written as minimising:

$$\int q(\boldsymbol{H}) \log q(\boldsymbol{H}) d\boldsymbol{H} - \int q(\boldsymbol{H}) \log p(\boldsymbol{H}) d\boldsymbol{H} + \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} \Bigg[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) + \hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \Big\} \Bigg]_+ \tag{5.54}$$

where the first term is the negative of the Shannon entropy. Since $q(\boldsymbol{H}) = \delta(\boldsymbol{H}, \hat{\boldsymbol{H}})$ is a Dirac delta function, this entropy is a constant but infinite value. Then the large margin training criterion (5.54) can be simplified to be the same as the training criterion for large margin training of the experts described in (E.14):

$$- \log p(\hat{\boldsymbol{H}}) + \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} \Bigg[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \Big(\hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) - \hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \Big) \Big\} \Bigg]_+ \tag{5.55}$$

Given the sampled indicators $\{\boldsymbol{z}^{(k)}\}_{k=1}^{K}$, The number of the represented experts can be determined: $M = \left| \{\boldsymbol{z}^{(k)}\}_{k=1}^{K} \right|$, which is the number of unique values in set $\{\boldsymbol{z}^{(k)}\}_{k=1}^{K}$. The prior distribution of $\boldsymbol{H}$ is assumed to be $p(\boldsymbol{H}) = \prod_{m=1}^{\infty} p(\boldsymbol{\eta}_m)$. For the unrepresented experts (having index $m > M$), minimising criterion (5.55) yields the mode of the prior distribution $p(\boldsymbol{\eta}_m)$. For the represented experts (having index $m \leq M$) with parameters $\hat{\boldsymbol{H}}_r = \{\hat{\boldsymbol{\eta}}_m\}_{m=1}^{M}$, analogous to the criterion (E.15) for large margin training of the experts, the large margin training criterion (5.55) can be described as the following M minimisation criteria:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{H}}_r) = \sum_{m=1}^{M} \mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) \tag{5.56}$$

where:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) = -\log p(\hat{\boldsymbol{\eta}}_m) + \frac{1}{K}\sum_{k=1}^{K}\sum_{\forall z_n^{(k)}=m,\forall n}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\mathcal{L}(W,W_n) - \hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\boldsymbol{\rho}_n) + \hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\boldsymbol{\rho})\Big\}\Bigg]_+ \tag{5.57}$$

In this work, the prior distribution of each expert's parameter is assumed to be a Gaussian distribution $p(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\eta}_m;\boldsymbol{\mu}_\eta,\Sigma_\eta)$ with mean $\boldsymbol{\mu}_\eta$ and a scaled identity matrix covariance $\Sigma_\eta = C\boldsymbol{I}$. In order to train the experts with limited training data and ensure good generalisation, the mean $\boldsymbol{\mu}_\eta$ is set to be the optimised parameter of the structured SVM trained with all the training data. Substituting the definition of the prior distribution $p(\boldsymbol{\eta}_m)$ in, the mth criterion can be further written as minimising:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) \propto \frac{1}{2}||\hat{\boldsymbol{\eta}}_m - \boldsymbol{\mu}_\eta||^2 + \frac{C}{K}\sum_{k=1}^{K}\sum_{\forall z_n^{(k)}=m,\forall n}\Bigg[\max_{W,\boldsymbol{\rho}\neq W_n,\boldsymbol{\rho}_n}\Big\{\hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\boldsymbol{\rho}) + \mathcal{L}(W,W_n)\Big\} - \hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\boldsymbol{\rho}_n)\Bigg]_+ \tag{5.58}$$

This criterion (5.58) is the same as the large margin training criterion (E.17) for training the experts of the infinite log-linear model. In this large margin training criterion (5.58), the bound of summation is for all $k \in \{1,\ldots,K\}$. As discussed in section E.2, this means the training data are replicated K times. For the nth training instance, it might be allocated to the same expert m for different k. Thus, the most competing hypothesis W can be cached, and the cached hypothesis can be reused when this training instance is allocated to the same expert again.

5.4.2.2 *The Estimation of $\hat{q}(\boldsymbol{\Theta},\boldsymbol{z})$*

Given the optimal distribution of the experts' parameters $\hat{q}(\boldsymbol{H}) = \delta(\boldsymbol{H},\hat{\boldsymbol{H}})$ defined in (5.53), the general criterion (5.44) with the combined criterion function (5.45) can be written as follows:

$$\begin{aligned}&\arg\min_{q(\boldsymbol{\Theta},\boldsymbol{z})}\left\{\text{KL}\big(q(\boldsymbol{\Theta},\boldsymbol{z})||p(\boldsymbol{\Theta},\boldsymbol{z})\big) - \int\sum_{\boldsymbol{z}}\mathcal{F}(\boldsymbol{\Theta},\boldsymbol{z};\mathcal{D})q(\boldsymbol{\Theta},\boldsymbol{z})d\boldsymbol{\Theta}\right\}\\ &\text{s.t.}\quad q(\boldsymbol{\Theta},\boldsymbol{z}) \in \mathcal{P}_{\text{prob}}\end{aligned} \tag{5.59}$$

where

$$\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{z}; \mathcal{D}) = \sum_{n} \Bigg(\log p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n) - \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{O}_n, \hat{\boldsymbol{H}}, z_n) \Big\} \Big]_+ \Bigg) \tag{5.60}$$

As discussed in section 5.2.1, the solution that minimises the general criterion is proportional to the product of the prior and the exponential of the criterion function. Thus, minimising criterion (5.59) yields:

$$\begin{aligned} \hat{q}(\boldsymbol{\Theta}, \boldsymbol{z}) &\propto p(\boldsymbol{\Theta}, \boldsymbol{z}) \exp(\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{z}; \mathcal{D})) \\ &= p(\boldsymbol{\Theta}, \boldsymbol{z}) \prod_{n=1}^{N} p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n) \cdot \\ &\quad \exp\Bigg(- \Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{ \mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{O}_n, \hat{\boldsymbol{H}}, z_n) \Big\} \Big]_+ \Bigg) \end{aligned} \tag{5.61}$$

As discussed in section 5.3.4, in Bayesian inference of the infinite log-linear model discussed, the posterior distribution of the model parameters obtained from Bayes' rule is equivalent to the solution that minimises the general criterion with the log-likelihood criterion function described in (5.41):

$$\begin{aligned} \hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) &\propto p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \exp(\mathcal{F}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}; \mathcal{D})) \\ &= p(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \prod_{n=1}^{N} p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n) P(W_n|\boldsymbol{O}_n, \boldsymbol{H}, z_n) \end{aligned} \tag{5.62}$$

In Bayesian inference of the infinite log-linear model, for the posterior distribution (5.62), the probability contributed from the expert is $P(W_n|\boldsymbol{O}_n, \boldsymbol{H}, z_n)$, which is the conditional distribution given by the log-linear model described in (5.24). By contrast, for the optimal distribution (5.61) in large margin training of the infinite log-linear model, the probability contributed from the expert is the exponential of the hinge loss function. This results from the hinge loss function in the definition of the combined criterion function defined in (5.45).

Since the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ described in (5.61) is intractable, Gibbs sampling is applied to sample from this distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, and the conditional distribution of each parameter will be discussed in detail in the rest of this section.

The Conditional Distribution of $\boldsymbol{\theta}_m$ Given the prior distribution $p(\boldsymbol{\Theta}, \boldsymbol{z}) = p(\boldsymbol{\Theta})p(\boldsymbol{z})$ and the sampled indicators $\boldsymbol{z}^{(k)} = \{z_1^{(k)}, \ldots, z_N^{(k)}\}$, the conditional distribution $\hat{q}(\boldsymbol{\Theta}|\boldsymbol{z}^{(k)})$ can be derived from the joint distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ described in (5.61):

$$\hat{q}(\boldsymbol{\Theta}|\boldsymbol{z}^{(k)}) \propto p(\boldsymbol{\Theta}) \prod_{n=1}^{N} p(\boldsymbol{O}_n|\boldsymbol{\Theta}, z_n^{(k)}) \tag{5.63}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ are the parameters of the Gaussian components[1], and $p(\boldsymbol{\Theta}) = \prod_m p(\boldsymbol{\theta}_m)$ is the prior distribution. Given the sampled indicators $\boldsymbol{z}^{(k)}$, the parameter of each component $\boldsymbol{\theta}_m$ is conditional independent. For the represented components (that have associated data), the conditional distribution of each parameter $\boldsymbol{\theta}_m$ can be written as:

$$\hat{q}(\boldsymbol{\theta}_m|\boldsymbol{z}^{(k)}) \propto p(\boldsymbol{\theta}_m) \prod_{\forall z_n^{(k)}=m} p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m\big) \tag{5.64}$$

This conditional distribution (5.64) is the same as the conditional posterior distribution of $\boldsymbol{\theta}_m$ (5.32) in Bayesian inference of the infinite log-linear model as discussed in section 5.3.1.4. The difference between the joint distributions for large margin training and Bayesian inference of the infinite log-linear model, described in (5.61) and (5.62) respectively, is the term corresponding to the experts. Since the terms corresponding to the experts (the last terms) in the joint distributions (5.61) and (5.62) are not functions of $\boldsymbol{\theta}_m$, the resulting conditional distributions of $\boldsymbol{\theta}_m$ from these joint distributions (5.61) and (5.62) are identical.

The Conditional Distribution of z_n Given the sampled parameters of the gating network $\boldsymbol{\Theta}^{(k-1)}$, according to the optimal joint distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ described in (5.61), the conditional distribution of the nth indicator variable z_n can be derived:

$$\begin{aligned}\hat{q}(z_n = m|\boldsymbol{z}_{-n}^{(k)}) \propto P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha)p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big)\cdot \\ \exp\Big(-\Big[\max_{W, \boldsymbol{\rho} \neq W_n, \boldsymbol{\rho}_n} \Big\{\mathcal{L}(W, W_n) - \mathcal{M}(W, W_n; \boldsymbol{O}_n, \hat{\boldsymbol{H}}, z_n)\Big\}\Big]_+\Big)\end{aligned} \tag{5.65}$$

where the set $\boldsymbol{z}_{-n}^{(k)}$ denotes all the indicators except $z_n^{(k)}$, namely $\boldsymbol{z}_{-n}^{(k)} = \{z_1^{(k)}, \ldots, z_{n-1}^{(k)}, z_{n+1}^{(k-1)}, \ldots, z_N^{(k-1)}\}$. In (5.65), the first term $P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha)$ is given by the Chinese restaurant process described in (5.31), and the second term $p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big)$ is the component likelihood, which is given by the Gaussian distribution.

[1] Again, the gating network is an infinite GMM.

In Bayesian inference of the infinite log-linear model discussed in section 5.3.1, the conditional posterior distribution of the indicator variable is described in (5.30):

$$\hat{q}(z_n = m|\boldsymbol{z}_{-n}^{(k)}) \propto P(z_n = m|\boldsymbol{z}_{-n}^{(k)}, \alpha)p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}_m^{(k-1)}\big)P(W_n|\boldsymbol{O}_n, \boldsymbol{\eta}_m^{(k-1)}) \tag{5.66}$$

The difference between these two conditional distributions (5.65) and (5.66) is the third term corresponding to the expert. This results from the different definitions of the criterion function given in (5.45) and (5.42), in which the hinge loss and the log-likelihood are used respectively.

When the indicator variable z_n indicates to an existing expert (the represented expert), the conditional posterior distribution of z_n can be obtained from (5.65) directly. When the indicator variable z_n indicates to a new expert (the unrepresented expert), similar to the method used in Bayesian inference of the infinite log-linear model model discussed in section 5.3.1, in calculating the likelihood $p\big(\varphi(\boldsymbol{O}_n)|\boldsymbol{\theta}\big)$, the parameter $\boldsymbol{\theta}$ is sampled from its prior distribution [131]. In order to ensure the newly generated expert has good generalisation, the parameter of the expert $\boldsymbol{\eta}$ is set to be the optimised parameter of the structured SVM trained with the whole training set.

5.4.3 *Infinite Structured SVMs*

In standard Bayesian inference, the model itself is probabilistic. A prior distribution is placed on the model parameters to express the uncertainty, and posterior distribution is obtained according to Bayes' rule, in which the (conditional) likelihood of the model is a probability. In section 5.4.1, an overall criterion is used in training the whole infinite log-linear model. Since training is a minimisation problem which does not require the model to be probabilistic, then discriminant functions (structured SVMs) can be employed as the experts of the infinite mixture of experts. This type of model is called the *infinite structured SVM* in this work, and will be discussed in the rest of this section.

A *discriminant function* is a function that maps the input $\boldsymbol{O}$ directly to a class W by choosing the class that maximises this function. Let the experts of the infinite log-linear model be discriminant functions, then the discriminant function for the zth expert can be described as:

$$F(W; \boldsymbol{O}, \boldsymbol{H}, z) = F(W; \boldsymbol{O}, \boldsymbol{\eta}_z) = \boldsymbol{\eta}_z^{\mathrm{T}}\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) \tag{5.67}$$

where $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of all the experts, and $\boldsymbol{\eta}_m$ is the parameter of the mth. $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$ is the joint feature vector, which is discussed in detail in section 3.5. Similar to the conditional probability of the infinite structured discriminative model (5.20), the overall discriminant function of the infinite structured SVM can be described as:

$$F(W; \boldsymbol{O}, \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}) = \sum_{z} F(W; \boldsymbol{O}, \boldsymbol{H}, z) P(z|\boldsymbol{O}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \tag{5.68}$$

For the infinite structured SVM, the training criterion is still the general criterion (for the infinite log-linear model) described in (5.44) with the combined criterion function (5.45), but the margin is defined as the difference of the discriminant functions:

$$\mathcal{M}(W_n, W; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = F(W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) - F(W; \boldsymbol{O}_n, \boldsymbol{H}, z_n) \tag{5.69}$$

Substituting the definition of the discriminant function (5.67) in, this margin (5.69) can be written as:

$$\mathcal{M}(W, W_n; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = \boldsymbol{\eta}_{z_n}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n) - \boldsymbol{\eta}_{z_n}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \boldsymbol{\rho}) \tag{5.70}$$

This margin (5.70) is identical to the margin (5.51) for the infinite log-linear model, where the margin is defined as the log ratio of the log-linear models as described in (5.46). The denominator terms of the log-linear models can be cancelled out, and the resulting margin becomes a linear function of $\boldsymbol{\eta}_{z_n}$ that is the same as the margin in (5.70) for the infinite structured SVM. This means large margin training of the infinite log-linear model and the infinite structured SVM are identical. Thus, the training process for the infinite structured SVM is the same as that for the infinite log-linear model discussed in section 5.4.2.

5.4.4 Classification

The (large margin) training process of the infinite log-linear model is summarised in Algorithm 3. After converging, the optimal distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \approx \hat{q}(\boldsymbol{H})\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ can be obtained, where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ are the parameters of the Gaussian components, $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of the experts, and $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ are the indicator variables corresponding to the training data. Since both the model discussed in this section and the one discussed in Appendix E are infinite log-linear models (but have different training approaches), given the optimal distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})$, in classification the class posterior distribution for the infinite log-linear model (with large training of the whole model)

is the same as that for the infinite log-linear (with large training of the experts) described in (E.22):

$$P(W|\boldsymbol{O},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} P(W|\boldsymbol{O},\hat{\boldsymbol{H}},z)P(z|\boldsymbol{O},\boldsymbol{\Theta}^{(k)},\boldsymbol{z}^{(k)}) \tag{5.71}$$

where z is the indicator variable corresponding to the input $\boldsymbol{O}$, and $\hat{\boldsymbol{H}}$ are the parameters of $\hat{q}(\boldsymbol{H})$, which is a Dirac delta function. M_k is the number of the represented experts, namely $M_k = |\boldsymbol{z}^{(k)}|$ which is the number of the unique values in set $\boldsymbol{z}^{(k)}$. $P(W|\boldsymbol{O},\hat{\boldsymbol{H}},z)$ is the conditional likelihood of the zth expert described in (5.24), and $P(z|\boldsymbol{O},\boldsymbol{\Theta}^{(k)},\boldsymbol{z}^{(k)})$ is the component posterior distribution as described in (E.21):

$$P(z|\boldsymbol{O},\boldsymbol{\Theta}^{(k)},\boldsymbol{z}^{(k)}) = \frac{\pi_z^{(k)}\mathcal{N}\big(\varphi(\boldsymbol{O});\boldsymbol{\theta}_z^{(k)}\big)}{\sum_z \pi_z^{(k)}\mathcal{N}\big(\varphi(\boldsymbol{O});\boldsymbol{\theta}_z^{(k)}\big)}, \quad z\in\{1,\ldots,M_k\} \tag{5.72}$$

In this component posterior distribution $\pi_z = N_z^{(k)}/N$, where $N_z^{(k)}$ is the number of training data associated with the zth expert. $\mathcal{N}(\cdot)$ is a Gaussian distribution.

Infinite Structured SVMs For the infinite structure SVM discussed in section 5.4.3, each expert is a discriminant function as described in (5.67). In classification, the overall discriminant function has a similar form to the class posterior distribution described in (5.71):

$$F(W;\boldsymbol{O},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} F(W;\boldsymbol{O},\hat{\boldsymbol{H}},z)P(z|\boldsymbol{O},\boldsymbol{\Theta}^{(k)},\boldsymbol{z}^{(k)}) \tag{5.73}$$

where $F(W;\boldsymbol{O},\hat{\boldsymbol{H}},z)$ is the discriminant function for the zth expert described in (5.67), and $P(z|\boldsymbol{O},\boldsymbol{\Theta}^{(k)},\boldsymbol{z}^{(k)})$ is the component posterior distribution described in (5.72). Compared with infinite log-linear models, classification for infinite structured SVMs is more efficient, given that the discriminant function is a linear function of the expert's parameter $\boldsymbol{\eta}_z$, and it does not have a normalisation term as that in the log-linear model.

5.5 Summary

In this chapter, a different perspective on Bayesian inference is introduced in section 5.1, where Bayesian inference is interpreted as a minimisation criterion. Section 5.2 introduces

the general criterion. The minimisation criterion for Bayesian inference is the general criterion with a log-likelihood criterion function. With different definitions of the criterion function, various training criteria can be resulted. Section 5.3 discusses the application of the infinite structured discriminative model to speech recognition. Standard Bayesian inference of the infinite log-linear model is detailed in section 5.3.1, and two approximation methods with MAP estimation and large margin training are discussed in section 5.3.2 and 5.3.3 respectively. In these approximations, the parameter of each expert is estimated by the MAP estimator and large margin training criterion respectively, replacing the process of sampling the parameter of each expert. Training of the infinite log-linear model with the general criterion having different criterion functions is also detailed. Large margin training of each expert (in a Gibbs sampling style training process discussed in section 5.3.3) is a special example of training all the experts with the general criterion having a hinge loss criterion function (discussed in Appendix E). Section 5.4 discusses the application of large margin training, in which the criterion is the general criterion with the combined criterion function, to the whole infinite log-linear model, the gating network and experts.

Chapter 6

Experiments

In the previous chapters, the log-linear model and the infinite log-linear model with various training approaches were detailed. The experiments of these models carried on different corpora (i.e. AURORA 2, AURORA 4 and the Babel corpora) will be studied in this chapter. On these data sets, the performances of the baseline systems, the log-linear models and the infinite log-linear model will be compared. In all experiments, the baseline HMM systems, i.e. tandem, hybrid and VTS compensated ones, are taken from various projects which involve a group of people working on. All the tandem, hybrid and joint decoding systems used in experiments have the state-of-the-art performance.

AURORA 2 is a noise-corrupted continuous digit corpus with vocabulary size 12 (one to nine, plus zero, oh and silence). On this small vocabulary task, some preliminary experiments were carried, e.g. the unstructured discriminative models and the infinite structured discriminative models discussed in section 5.3. AURORA 4 is a noise-corrupted medium to large vocabulary database based on the Wall Street Journal (WSJ) data. It is impractical to apply the unstructured discriminative model to this task. Thus, only the structured discriminative models and their infinite counterparts were examined on AURORA 4 set. Moreover, in order to make training of the models with large margin training criteria more efficient (or practical), the constraint set propagation approach was also applied. In order to examine the performances of the structured discriminative models (the log-linear model and structured SVM) and the infinite structured discriminative models described

in section 5.3 on real world data, experiments were also conduced on the Babel corpora, which cover a range of diverse languages and are recorded in real-life scenarios, such as conversational telephone speech, over a range of acoustic conditions, such as mobile phone conversation made from car [64].

6.1 Experiments on AURORA 2

In this section, experiments conducted on the AURORA 2 corpus will be discussed. Since AURORA 2 is a small vocabulary noise corrupted continuous digit data set, the unstructured discriminative model, such as the SVM, can be implemented. Comparisons of the performances will be made between the unstructured discriminative models and the structured discriminative models. This section will also give the performances of the infinite structured discriminative models described in section 5.3 on the AURORA 2 corpus. In order to make training more efficient, constraint set propagation was applied in training the infinite structured discriminative model described in section 5.3.3. Efficiency comparison will also be made between the infinite models with and without constraint set propagation.

6.1.1 The AURORA 2 Corpus

The AURORA 2 corpus [137] is designed to evaluate the performance of speech recognition algorithms in various noisy conditions. AURORA 2 is based on the TIDigits database [111] with noise artificially added, and the vocabulary size is 12 (from "one" to "nine", plus "zero", "oh" and "silence"). In this database, 8 real-world noises are added to the utterances over a variety of signal to noise ratios (SNRs).

There are two types of training data in the AURORA 2 database. The first is the clean training data with 8840 utterances recorded from 55 male and 55 female adults. The second is the multi-condition training data with 8840 utterances, which are the clean training data corrupted using different noises and SNRs. The multi-condition training data are divided into 20 subsets with 422 utterances each, according to 4 noises (N1, N2, N3 and N4) and 5 SNRs (20dB, 15dB, 10dB, 5dB and the clean speech). These 4 noises are suburban train, babble, car and exhibition hall respectively.

For the test data in AURORA 2, 4004 utterances recorded from 52 male and 52 female speakers are divided into 4 subsets with 1001 utterances in each. There are 3 test sets called test set A, test set B and test set C, and they are comprised of these 4 subsets. Test set A consists of the 1st subset corrupted by 4 noises (N1, N2, N3 and N4) and 7 SNRs (20dB, 15dB, 10dB, 5dB, 0dB, -5dB and the clean speech). These 4 noises are the same as the noises in the multi-condition training data, namely suburban train, babble, car and exhibition hall. Test set B consists of the 2nd subset corrupted by 4 noises (N5, N6, N7 and N8) and 7 SNRs. These 4 noises are restaurant, street, airport and train station respectively. Test set C consists of the 3rd and 4th subsets. Each subset is corrupted with channel distortion by one type of noise and 7 SNRs. These noises (N9 and N10) are suburban train and street respectively. In this thesis, a subset of the test data is used. All the experiments are conduced with test sets having SNRs 20dB, 15dB, 10dB, 5dB and 0dB. Thus, the number of utterances in test set A, B and C are 20020, 20020 and 10010 respectively.

6.1.2 *Experiments Setup*

The baseline system used in the experiments is the vector Taylor series (VTS) compensated HMMs. The clean trained HMMs are trained on the clean data with 8840 utterances recorded from 55 male and 55 female adults. The feature vectors used by the HMMs have 39 dimensions. These features consist of 12 MFCCs appended with zeroth cepstrum, and delta and delta-delta coefficients are used. The HMMs are 16 emitting state whole word digit models, and the output distribution of each state is a GMM with 3 mixtures and diagonal covariance matrices. Because of the mismatch between the clean training data and the noise corrupted test data, the VTS model-based compensation [2] described in section 2.4.3 is applied. In the VTS compensation, the noise model parameters for each utterance are estimated based on the Maximum Likelihood (ML) noise estimation scheme [63, 115]. The word error rate (WER) of the clean trained HMMs and VTS-compensated HMMs is listed in Table 6.1.

In all the experiments (except the clean trained and VTS-compensated HMMs), a subset of the multi-condition training data is used. The training data contains 4 noises (N2, N3 and N4) and 3 SNRs (20dB, 15dB and 10dB). The test data is also a subset, which only contains the sets having SNRs 20dB, 15dB, 10dB, 5dB and 0dB.

System	Test Set WER(%)			Avg (%)
	A	B	C	
clean HMM	43.90	46.60	35.70	43.30
VTS-HMM	9.84	9.11	9.53	9.48

Table 6.1: *The performance of the clean trained HMMs and the VTS-compensated HMMs on the AURORA 2 corpus.*

6.1.3 The Infinite GMM

In this subsection, some preliminary experiments with the infinite GMMs (iGMMs), which were introduced in section 4.3.4, will be discussed. The iGMMs can be used as classifiers in an acoustic code-breaking fashion (discussed in section 3.1.3), where utterances are segmented into segments, and each segment is treated independently and classified separately. The iGMM cannot be directly applied to the temporal sequence data (even for segments). In to handle the dynamic aspect of the data, the features (or scores) based on generative models were used. By using this type of features, the noise robust technologies used by the generative models can also be employed in generating the features. In this work, these features are log-likelihood features generated by the VTS compensated HMM system, which was discussed in section 2.4.3 and has the same configuration as [63]. This type of features was discussed in detail in section 3.5.2.1.

Given the segmentation (from the VTS compensated HMM system), the utterance $\boldsymbol{O}$ can described as $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \cdots, \boldsymbol{O}_{(I)}\}$, where I is the number of segments, and each segment $\boldsymbol{O}_{(i)}$ is one word (or silence). As defined in (3.69), the log-likelihood features for segment $\boldsymbol{O}_{(i)}$ can be expressed as follows:

$$\varphi\left(\boldsymbol{O}_{(i)}\right) = \begin{bmatrix} \log p\left(\boldsymbol{O}_{(i)}|v_1\right) \\ \vdots \\ \log p\left(\boldsymbol{O}_{(i)}|v_L\right) \end{bmatrix} \tag{6.1}$$

where $\{v_1, \cdots, v_L\}$ are the whole vocabularies with size L, and $p\left(\boldsymbol{O}_{(i)}|v_l\right)$ is the likelihood for segment $\boldsymbol{O}_{(i)}$ associated with word v_l. Based on the log likelihood features, the features

used by the iGMM are normalised, which are the "log-posterior" features[1]:

$$\boldsymbol{x}_i = \varphi'(\boldsymbol{O}_{(i)}) = \log \frac{\exp\big(a\varphi(\boldsymbol{O}_{(i)})\big)}{\mathrm{sum}\big(\exp(a\varphi(\boldsymbol{O}_{(i)}))\big)} \tag{6.2}$$

where $\exp(\cdot)$ is the element-wise exponential function performed to all elements of the vector, and $\mathrm{sum}(A)$ denotes the summation of elements in A. The "acoustic deweighting" factor a is empirically chosen as 0.1 in this work.

The iGMM is a generative model, in classification the posterior distribution of the class label (word), which can be derived according to Bayes' rule, is used:

$$\hat{w} = \arg\max_w P(w|\boldsymbol{x}, \mathcal{D}_w) = \arg\max_w P(w)p(\boldsymbol{x}|\mathcal{D}_w) \tag{6.3}$$

where w is the word to be estimated and $w \in \{v_1, \ldots, v_L\}$. $P(w)$ is the prior of the word (language model), which is considered as a uniform distribution for all digits in this work, namely $P(w) = 1/L$. $p(\boldsymbol{x}|\mathcal{D}_w)$ is the the likelihood given by the iGMM associated with word w. In this likelihood, all model parameters $\mathcal{G}_w$ are marginalised out. Since the marginalisation is intractable for the infinite GMM, the integral (over model parameters) is approximated by summing over K samples:

$$p(\boldsymbol{x}|\mathcal{D}_w) = \int p(\boldsymbol{x}|\mathcal{G}_w)p(\mathcal{G}_w|\mathcal{D}_w)d\mathcal{G}_w \approx \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{G}_w^{(k)}) \tag{6.4}$$

where $\mathcal{G}_w^{(k)} \sim p(\mathcal{G}_w|\mathcal{D}_w)$, and $p(\mathcal{G}_w|\mathcal{D}_w)$ is the joint posterior distribution of the model parameters for the iGMM. Since this joint posterior distribution is highly complicated, here Gibbs sampling is used to sample these parameters [131, 149]. In classification, similar to the infinite log-linear model discussed in section 5.3.1.1, only represented components (which are the Gaussian components that have associated data) are considered. Thus, the likelihood $p(\boldsymbol{x}|\mathcal{G}_w^{(k)})$ is given by a GMM and $\mathcal{G}_w^{(k)}$ are the corresponding model parameters.

By substituting the likelihood (6.4) and $P(w) = 1/L$ into equation (6.3), classification with iGMM can be expressed as follows:

$$\hat{w} \approx \arg\max_w \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{G}_w^{(k)}) \tag{6.5}$$

[1] In order to simplify the notation, the feature vector for an segment is denoted as $\boldsymbol{x}$ here.

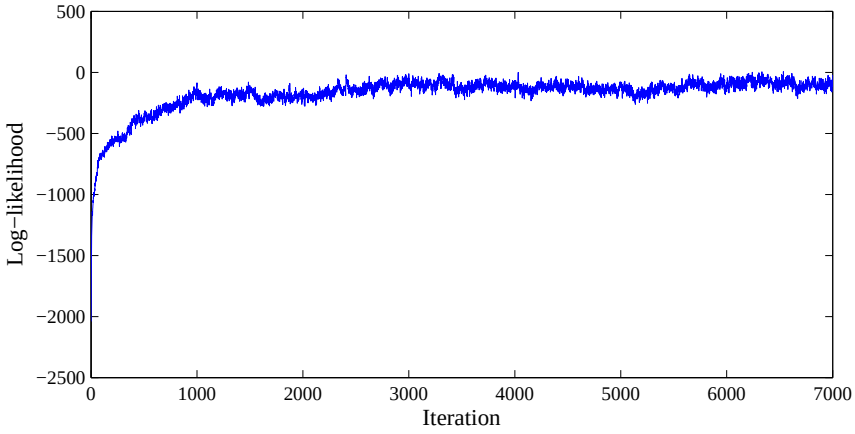


Figure 6.1: *The change of log-likelihood*

6.1.3.1 Convergence and Correlation

In inference (Gibbs sampling) of the iGMM, the samples $\{\mathcal{G}_w^{(k)}, \ldots, \mathcal{G}_w^{(K)}\}$ are aimed to be drawn from the true posterior distribution of the parameters $p(\mathcal{G}_w|\mathcal{D}_w)$ and they are not correlated to each other. However, neither of these is true [190]. In Gibbs sampling, it takes a long time to converge and consecutive samples are positively correlated with each other. Thus, before using these samples, convergence of the Markov chain[1] need to be detected. After converging, only part of the samples are used to reduce the correlation among samples[2].

In order to detect whether the sampler is converged, a number of formal tests, such as the Geweke test [71], can be used. In the test, after removing the samples drawn from the burn-in period[3], two sets of samples can be obtained from the remaining samples, i.e. the first 10% and the last 50% of the samples. If the sampler converges, the means of these two sets should be equal. Empirically, this formal test is too strict and rarely can be satisfied, hence the log-likelihood is used to diagnose convergence. Figure 6.1 illustrates the log-likelihood given by the iGMM in different iterations of Gibbs sampling. According the

[1] Gibbs sampling is one form of the MCMC methods, which are a class of algorithms for sampling from a probability distribution based on a Markov chain whose equilibrium distribution is the desired distribution.

[2] In order to reduce computation, fewer number of independent samples are preferred, rather than a large amount of correlated ones.

[3] In the burn-in period, the sampler is not converged, and the samples drawn during this period bias to the starting point.

System	Train Adapt	Test Adapt	Test Set WER (%)			Avg (%)
			A	B	C	
VTS-HMM	–	–	9.84	9.11	9.53	9.48
iGMM	–	–	11.47	10.95	11.63	11.30
	–	CMLLR	9.49	9.38	9.54	9.46
	CMLLR	CMLLR	9.62	9.67	9.72	9.62

Table 6.2: *The performance of the iGMM on the AURORA 2 corpus.*

figure, the first 3000 samples can be considered as in the burn-in period and they will be discarded.

The adjacent samples from Gibbs sampler are positively correlated [190], and independent samples are preferred. In order to reducing the correlation among samples, only a part of the samples are used by storing every ith sample ($i \geq 1$) and discarding the others. This process is also known as *thinning*. The effective sample size gives an estimate of the equivalent number of independent data from the correlated samples, and it can be expressed as follows [123]:

$$N_{\mathrm{e}} = N\frac{1-r_1}{1+r_1} \tag{6.6}$$

where N_{e} is the *effective sample size*, N the size (number) of correlated samples, and r_1 is the first order autocorrelation coefficient. The ratio $\frac{1-r_1}{1+r_1}$ is a scaling factor multiplied by the original sample size to compute the effective sample size [123]. The autocorrelation coefficient r_1 is defined as follows:

$$r_1 = \frac{\sum_{n=1}^{N-1}(\theta_n - \bar{\theta})(\theta_{n+1} - \bar{\theta})}{\sum_{n=1}^{N}(\theta_n - \bar{\theta})^2} \tag{6.7}$$

where θ_n is one sampled parameter, such as the number of represented components for the iGMM. $\bar{\theta} = \frac{1}{N}\sum_{n=1}^{N}\theta_n$ is the mean. In the convergence test, for example, if $r_1 = 0.974$, then the scaling factor is 0.013, and these samples can be thinned by storing every 77th sample ($\frac{1}{0.013} \approx 77$). On the AURORA 2 corpus, empirically, every 100th sample is stored in the thinning process.

6.1.3.2 *Preliminary Experiments*

Given the sampled parameters, the iGMM can be viewed as a "gigantic" GMM which consists of K GMMs as described at the right hand side of expression (6.4). This allows standard adaptation approaches (discussed in section 2.4) to be applied to the iGMM. In this work, global CMLLR[1] [55] was used. As discussed in section 2.4.2.2, CMLLR can be operated in the form of transforming the input features. Thus, the sampling process for iGMM does not need to be changed, but using the transformed features.

In experiments, both adaptation and adaptive training (discussed in section 2.4.4) were implemented, and diagonal covariance matrices were used for the iGMM. The performances of the iGMM with and without adaption are tabulated in the second block of Table 6.2. In this block, the first row gives results for the iGMM without adaptation; For the second row, adaptation was applied to the test data, where the CMLLR transform was estimated for each noise condition and signal-to-noise ratio (SNR) separately; The third row uses both adaptation and adaptive training. In this work unsupervised adaptation was used. The labels of the test data were estimated by the baseline system, which is the VTS compensated HMM system [63], and the performance of this baseline is given in the first block of Table 6.2.

In all configurations, the iGMM has poor performance. Although by using adaptation and adaptive training, the word error rate (WER) can be reduced, the performance is still bad compared with the baseline VTS-HMM system. The iGMM is a generative model, and might be limited for a classification task using generative features. As reported in [63, 208], by using discriminative models based on the generative features, such as the log-likelihood features used by the iGMM, considerable performance gains can be achieved compared with the baseline VTS-HMM system. This motivates the application of discriminative models, which directly models the conditional distribution of the class given the input.

6.1.4 *The Unstructured Discriminative Models*

The previous subsection discussed the application of the iGMMs (generative models) to the AURORA 2 corpus. The iGMMs were used in an acoustic code-breaking fashion, where utterances are segmented into segments, and each segment is treated independently and clas-

[1] It is not possible to use standard regression classes, as the component number and parameters are not fixed. However, it is possible to use class-specific transforms.

System	Criterion	Features	Test Set WER (%)			Avg (%)
			A	B	C	
VTS-HMM	ML	MFCC	9.84	9.11	9.53	9.48
LLM	Large Margin	Log-like	8.29	7.90	8.61	8.20
iLLM	Large Margin*		8.25	7.87	8.53	8.15
LLM	Large Margin	Derivative	8.28	7.85	8.63	8.18
iLLM	Large Margin*		8.05	7.81	8.44	8.04

Table 6.3: *The performance of the unstructured discriminative models on the AURORA 2 corpus. The LLM with large margin training can be interpreted as a multi-class SVM. For the iLLM, large margin training is applied to each expert. This is an approximate approach. $*$ indicates approximations are made in Gibbs sampling, and "*Large Margin***" denotes sampling for each expert is replaced by large margin training.*

sified separately. For a classification task, discriminative models might be preferred [132]. Alternative to generative models, unstructured discriminative models will be examined in this subsection. For unstructured discriminative models, the structure of the label is not considered. Thus, classification of the whole sentence is impractical. Analogous to the iGMM discussed in the previous subsection, unstructured discriminative models can be implemented in an acoustic code-breaking fashion. Also, the segmentations are the most likely segmentations given by the VTS compensated HMM system.

In addition to the log-likelihood features, the derivative features described in section 3.5.2.1 were also used by the unstructured discriminative models. To keep training with derivative features feasible, only the first element of the derivative with respect to each mean were used. In this thesis, the unstructured discriminative models examined are the (unstructured) log-linear model[1] (LLM) discussed in section 3.1.1 and the infinite log-linear model[2] (iLLM) discussed in section 5.3. Large margin training of the (unstructured) log-linear model was studied, which can be interpreted as a multi-class SVM as discussed in section 3.1.2.1. For the iLLM, each expert of the infinite model were trained with large margin criterion as discussed in section 5.3.3. The results of these unstructured discriminative models are given in Table 6.3. In this table, for the iLMM, $*$ indicates approximations are

[1] It is also known as multinomial logistic regression.

[2] In this subsection, only unstructured models are considered, hence each expert (log-linear model) of the infinite model is multinomial logistic regress discussed in section 3.1.1.

made in Gibbs sampling, and "Large Margin*" denotes sampling for each expert is replaced by large margin training as discussed in section 5.3.3. For the iLLM, the class posterior distribution used in classification can be described in a same form as the infinite structured model (5.28):

$$P(w|\boldsymbol{O},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} P(w|\boldsymbol{O},\boldsymbol{\eta}_z^{(k)})P(z|\boldsymbol{O},\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)}) \tag{6.8}$$

where the probability $P(w|\boldsymbol{O},\boldsymbol{\eta}_z^{(k)})$ is given by the unstructured log-linear model (multinomial logistic regression) discussed in section 3.1.1, and $P(z|\boldsymbol{O},\boldsymbol{\pi}^{(k)},\boldsymbol{\Theta}^{(k)})$ is the component posterior distribution (5.21) given by the gating network. K is the number of samples used in classification, and M_k is the number of represented experts for the kth draw. In experiments, the number of samples K is 10.

As shown in Table 6.3, all discriminative models outperform the VTS compensated HMM baseline system. On the log-likelihood feature space and derivative feature space, the iLLMs have better performance than the LLMs. One possible reason for this gain is that the iLLM explores the distribution of the training data and infers the number of experts, then applies different experts focus on different regions of the feature space to make an ensemble decision, rather than using a single classifier on the whole feature space. Another reason might be the iLLM employs multiple experts to yield an overall non-linear decision boundary, rather than a linear one from the LLM. By using derivative features, more information, such as long-span dependences of observations [48], is provided, better performance can be achieved compared with using log-likelihood features.

For the iLLM, each expert was trained with the large margin training criterion, and the training criterion is the one for the multi-class SVM[1] as described in expression (3.25). Different experts of the iLLM share the same C (which is a trade-off between the regularisation term and the hinge loss), and in this work the optimal C was tuned on the test set A. Figure 6.2 illustrates the WER of the iLLM on different feature spaces with various C. Since the parameter of each expert is given a Gaussian prior with non-zero mean $\boldsymbol{\mu}_\eta$ (which is optimal parameter of the multi-class SVM trained with all training data), the iLLM only achieves the baseline performance of the multi-class SVM when the C is small, and the optimised configuration can be obtained by gradually increasing C. Thus, by introducing this

[1] Again, large margin training of the unstructured log-linear model (multinomial logistic regression) can be interpreted as a multi-class SVM as discussed in section 3.1.2.2.

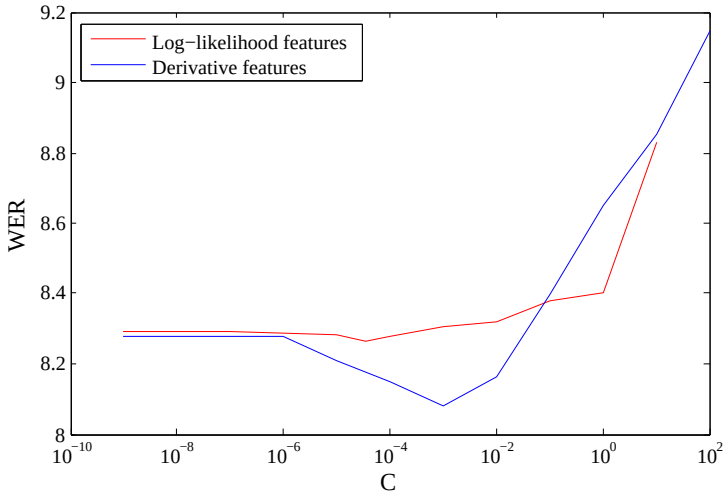


Figure 6.2: *The performance of the iLLM with large margin training of each expert on the test set A of the AURORA 2 corpus with different C.*

non-zero mean $\boldsymbol{\mu}_\eta$, the iLLM can at least achieve the performance of the multi-class SVM. Without this mean $\boldsymbol{\mu}_\eta$, the iLLM could have poor performance, as not all the experts have enough associated data.

In experiments, when comparing the performance of different algorithms, the variability and uncertainty of the algorithms need to be considered. For example, two algorithms have the same true error rate. They may give different error rate on a test set due to variability. In this thesis, the algorithms were tested on different corpora, and consistent gains have been observed, which will be shown in the following sections. More formally, a statistical hypothesis test, such as the McNemar's test or the matched-pairs test [73, 122], can be used to verify whether the performance difference of the algorithms is statistically significant. In statistical hypothesis testing, statistical significance is attained when the p-value[1] is less than the significance level[2]. The results are said to be statistically significant at given confidence level, when the computed confidence interval fails to contain the value specified by the null hypothesis [194].

[1] The p-value is the probability of obtaining at least as extreme results given that the null hypothesis is true, and it measures how extreme the observation is.

[2] The significance level is the probability of rejecting the null hypothesis given that it is true.

6.1.5 The Structured Discriminative Models

In the previous subsections, classification with the unstructured models are examined on the AURORA 2 corpus. In speech recognition, inputs are sequential data and class labels are sentences. In order to classify the whole utterance, the label structure needs to be considered. In the structured model, rather than treating each sentence as a single class, sentences can be broken into word (or sub-word) units. This section will discuss the performance of various structured discriminative models on the AURORA 2 corpus. The practical issues of large margin training the infinite structured discriminative model described in section 5.3.3 will also be discussed.

6.1.5.1 Constraint Set Propagation

As discussed in section 5.3.3, for the infinite log-linear model (iLLM), large margin training is applied to each expert which is a log-linear model. The training criterion for each expert is equivalent to the large margin criterion for a structured SVM. Thus, the efficient training approaches, such as the 1-slack cutting plane algorithm [96], for structured SVMs can also be applied when it comes to the iLLM (with large margin training of each expert). By using the 1-slack cutting plane algorithm, the number of constraints becomes much fewer [96], training thus becomes more efficient. The 1-slack cutting plane algorithm for the mth expert is detailed in Algorithm 4[1]. In this algorithm, the process of finding the best competing hypotheses described in equation (6.11) can be paralleled.

As described in the 1-slack cutting plan algorithm (Algorithm 4), initially the constraint set is empty. Rather than starting from an empty set, this initial constraint set can be generated or be modified from the constraint set in the last iteration of optimising that expert's parameter, and the detail is described in Algorithm 5.

In the 1-slack cutting plane algorithm, each iteration of solving the quadratic problem (6.9), the inequality constraints in (6.10) are applied. For the inequalities, each sum is treated as a whole. Thus, the form of the constraint set described in (6.12) can be expressed as:

$$\mathbb{W} \leftarrow \mathbb{W} \cup \{V; s\} \tag{6.13}$$

[1] The prior of the expert's parameter with zero mean is discussed here. The non-zero mean $\boldsymbol{\mu}_\eta$ will lead to a similar form [210].

Algorithm 4: The 1-slack cutting plane algorithm

input : $\{\boldsymbol{O}_n, W_n, \boldsymbol{\rho}_n\}_{\forall z_n=m}$, C and precision ϵ;

Initialisation: $\mathbb{W} \leftarrow \phi$

repeat

$$\{\boldsymbol{\eta}_m, \xi\} \leftarrow \arg\min_{\boldsymbol{\eta}_m, \xi \geq 0} \frac{1}{2}||\boldsymbol{\eta}_m||^2 + C\xi \tag{6.9}$$

$$\text{s.t. } \forall \text{ elements in } \mathbb{W}: \quad \boldsymbol{\eta}_m^{\mathsf{T}} \sum_{z_n=m} \Delta\Phi_n \geq \sum_{z_n=m} \mathcal{L}(\mathcal{W}, \mathcal{W}_n) - \xi \tag{6.10}$$

where $\Delta\Phi_n = \Phi(\mathcal{O}_n, \mathcal{W}_n; \boldsymbol{\rho}_n) - \Phi(\mathcal{O}_n, \mathcal{W}; \boldsymbol{\rho})$

for $\forall z_n = m$ **do** */* find the best competing hypotheses */*

$$W_n^*, \boldsymbol{\rho}_n^* \leftarrow \arg\max_{W, \boldsymbol{\rho}} \{\mathcal{L}(W, W_n) + \boldsymbol{\eta}_m^{\mathsf{T}} \Phi(\boldsymbol{O}, W; \boldsymbol{\rho})\} \tag{6.11}$$

end

$$\mathbb{W} \leftarrow \mathbb{W} \cup \{W_n^*, \boldsymbol{\rho}_n^*\}_{\forall z_n=m} \tag{6.12}$$

until */* no constraint can be found that is violated more than ϵ */*

$\forall$ elements in $\mathbb{W}$: $\sum_{z_n=m} \mathcal{L}(W, W_n) - \boldsymbol{\eta}_m^{\mathsf{T}} \sum_{z_n=m} \Delta\Phi_n \leq \xi + \epsilon$;

return: $\boldsymbol{\eta}_m$

In this constraint set definition (6.13), $V = \sum_{z_n=m} \left(\Phi(\boldsymbol{O}_n, W_n; \boldsymbol{\rho}_n) - \Phi(\boldsymbol{O}_n, W_n^*; \boldsymbol{\rho}_n^*)\right)$ is a vector, and $s = \sum_{z_n=m} \mathcal{L}(W_n^*, W_n)$ is a scalar. W_n^* and $\boldsymbol{\rho}_n^*$ are the best competing hypothesis and corresponding segmentation obtained according to (6.11) by given the current value of $\boldsymbol{\eta}_m$. Thus, once the values of $\boldsymbol{\eta}_m$ in different iterations of the cutting plane algorithm are given, the constraint set $\mathbb{W}$ can be determined. Thus, in order to generate set $\mathbb{W}$, a set $\mathbb{G}$ can be defined, which consists of the values of $\boldsymbol{\eta}_m$ in different iterations of the cutting plane algorithm (Algorithm 4). The set $\mathbb{G}$ can be obtained along with set $\mathbb{W}$, then expression (6.12), which defines $\mathbb{W}$, can be written as:

$$\mathbb{W} \leftarrow \mathbb{W} \cup \{V, s\} \; ; \quad \mathbb{G} \leftarrow \mathbb{G} \cup \boldsymbol{\eta}_m \tag{6.14}$$

For the iLLM (with large margin training of each expert), the current constraint set $\mathbb{W}$ cannot be directly used as the initial constraint set in the next iteration of training the expert, given that the assignment of the data to experts might be changed, namely the training data associate with one expert might be different. Although the initial constraint set can be generated according to (6.11) and (6.13) given $\mathbb{G}$, when the number of training data is large, this constraint set generalisation approach becomes quite inefficient. Thus, a more elegant

Algorithm 5: Modify the constraint set

input : $\mathbb{W}_m^{(k)} = \{V_j, s_j\}_{j=1}^J$, $\mathcal{A}_m^{(k)}$, $\mathcal{B}_m^{(k)}$, and $\mathbb{W}_m^{(k+1)} = \phi$;
for $\boldsymbol{\eta}_j \in \mathbb{G}_m^{(k)}$ **do**
  for $n \in \mathcal{A}_m^{(k)}$ **do** */* remove data from constraints */*
    $W_{j,n}^*, \boldsymbol{\rho}_{j,n}^* \leftarrow \arg\max_{W,\boldsymbol{\rho}} \{\mathcal{L}(W, W_n) + \boldsymbol{\eta}_j^\mathsf{T} \Phi(\boldsymbol{O}_n, W; \boldsymbol{\rho})\}$
    $V_j = V_j - \big(\Phi(\boldsymbol{O}_n, W_n; \boldsymbol{\rho}_n) - \Phi(\boldsymbol{O}_n, W_{j,n}^*; \boldsymbol{\rho}_{j,n}^*)\big)$
    $s_j = s_j - \mathcal{L}(W_{j,n}^*, W_n)$
  end
  for $n \in \mathcal{B}_m^{(k)}$ **do** */* add data to constraints */*
    $W_{j,n}^*, \boldsymbol{\rho}_{j,n}^* \leftarrow \arg\max_{W,\boldsymbol{\rho}} \{\mathcal{L}(W, W_n) + \boldsymbol{\eta}_j^\mathsf{T} \Phi(\boldsymbol{O}_n, W; \boldsymbol{\rho})\}$
    $V_j = V_j + \big(\Phi(\boldsymbol{O}_n, W_n; \boldsymbol{\rho}_n) - \Phi(\boldsymbol{O}_n, W_{j,n}^*; \boldsymbol{\rho}_{j,n}^*)\big)$
    $s_j = s_j + \mathcal{L}(W_{j,n}^*, W_n)$
  end
  $\mathbb{W}_m^{(k+1)} \leftarrow \mathbb{W}_m^{(k+1)} \cup \{V_j, s_j\}$
end
return: $\mathbb{W}_m^{(k+1)}$

and efficient constraint set modification method is applied to obtain the initial constraint set in this work.

Let $\mathcal{A}_m^{(k)}$ be the index of the data associated with the mth expert and to be allocated to other experts in the next iteration of training this expert, and $\mathcal{B}_m^{(k)}$ the index of the data associated with other experts and to be assigned to the mth expert in the next iteration. Given the current constraint set $\mathbb{W}_m^{(k)} = \{V_j, s_j\}_{j=1}^J$, the corresponding parameter set $\mathbb{G}_m^{(k)} = \{\boldsymbol{\eta}_j\}_{j=1}^J$, and index sets $\mathcal{A}_m^{(k)}$ and $\mathcal{B}_m^{(k)}$, the initial constraint set $\mathbb{W}_m^{(k+1)}$ in the next iteration of training the mth expert can be obtained according to Algorithm 5.

In Algorithm 5, the best competing hypothesis $W_{j,n}^*$ and the corresponding segmentation $\boldsymbol{\rho}_{j,n}^*$ can be found through Viterbi algorithm [210]. Let $|\mathcal{A}_m^{(k)}|$ be the number of items in set $|\mathcal{A}_m^{(k)}|$, and $|\mathcal{B}_m^{(k)}|$ the number of items in set $\mathcal{B}_m^{(k)}$. In order to obtain one constraint for every expert, $\sum_m (|\mathcal{A}_m^{(k)}| + |\mathcal{B}_m^{(k)}|)$ calls of the Viterbi algorithm are required to find the best competing hypotheses. Let $N_c^{(k)}$ denote the number of indicators whose sampled values are changed, since the change for one indicator affects the associated data of two experts, the number of calls for the Viterbi algorithm is twice the number of indicators whose sampled

C	Time (hour)		Avg. change of indicators	Avg. active constraint #
	without	with		
1e-06	1.68	1.45	1.66%	1.00
1e-05	2.31	1.47	1.67%	3.55
1e-04	5.81	1.58	1.70%	7.09
1e-03	14.48	3.26	1.67%	14.62
1e-02	>24	7.83	1.69%	29.08

Table 6.4: *Computational time of the iLLM (with large margin training of each expert) training with and without constraint set propagation on the AURORA 2 corpus.*

values are changed, namely $\sum_m(|\mathcal{A}_m^{(k)}| + |\mathcal{B}_m^{(k)}|) = 2N_c^{(k)}$. However, for the constraint set generalisation approach, the Viterbi algorithm needs to be applied N times, where N is the number of the whole training data. Normally, $2N_c^{(k)}$ is far smaller than N, so the constraint set modification method is much more efficient than the generalisation approach.

Table 6.4 describes the efficiency of iLLM (with large margin training of each expert) training with and without constraint set propagation in 1000 Gibbs sampling iterations with various C (having the same precision ϵ in the 1-slack cutting plane algorithm), and the constraint set modification approach described in Algorithm 5 was used.

Training of the iLLM can be divided into two parts: the gating network and the experts (LLMs). For different C, the time of training the gating network can be viewed as a constant, but the time of training the experts varies. When C is small, this results in a small number of constraints, and training of experts is very fast. Thus, the time of training the iLLM is dominated by training of the gating network. Thus, when C is small, the time of training the iLLM with and without constraint set propagation is similar as shown in Table 6.4. As C grows, the training time increases, but the iLLM trained with constraint set propagation is much more efficient. The reason is that, for the iLLM training with constraint set propagation, rather than starting from an empty initial constraint set, the active constraint set [181] (that affects the solution) in the last iteration of training the expert is modified to be the current initial constraint set. The computational time of obtaining the initial constraint set is determined by both the number of active constrains and the number of the indicators whose values are changed. As shown in Table 6.4, for various C, the average ratio of the changed indicators to all training data is very small, i.e. around 1.7%. Thus, the

System	Criterion	Test Set WER(%)			Avg.
		A	B	C	
VTS-HMM	ML	9.83	9.11	9.53	9.48
LLM	CML	8.21	7.74	8.36	8.05
	Large Margin	7.97	7.54	8.31	7.86
iLLM	Bayesian	8.19	7.71	8.36	8.03
	CML*	8.22	7.71	8.35	8.04
	Large Margin*	7.69	7.39	7.98	7.63

Table 6.5: *The performance of different systems on the AURORA 2 corpus. "Bayesian" denotes Bayesian inference of the iLLM, where Gibbs sampling is used. "$*$" indicates approximations are made in Gibbs sampling. "CML*" means sampling for each expert is replaced by CML training, and "Large Margin*" denotes sampling for each expert is replaced by large margin training.*

time of obtaining the initial constraint set is mainly determined by the number of the active constrains.

As shown in Table 6.4, by using constraint set propagation, the system efficiency can be improved significantly, especially when C is large. Since large margin training of the log-linear model (or structured SVM) is a convex optimisation problem, the final optimum will not be affected by the initial status. Thus, the performance of the systems with and without constraint set propagation should stay the same.

6.1.5.2 *Experimental Results*

In this subsection, the experimental results of the iLLMs discussed in section 5.3 will be presented, i.e. Bayesian inference of the iLLM, and two approximate approaches. In these two approximate methods, conditional maximum likelihood (CML) estimation and large margin training were used to replace sampling for each expert. The experimental results of these approaches are given in Table 6.5, in which the performance of the baseline parametric models are also given. In the top block are the baseline numbers: the VTS compensated HMM, and the single log-linear model (LLM) using the log-likelihood features from that HMM, trained probabilistically or with a large-margin criterion. The bottom block has infinite models. The row labelled "Bayesian" is Bayesian inference of the iLLM discussed

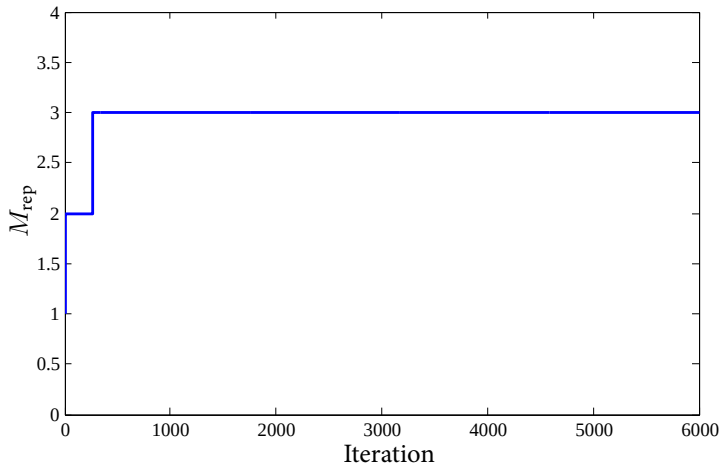


Figure 6.3: *The change of the number of represented experts.*

in section 5.3.1, where Gibbs sampling is used. This approach does not yield a significant improvement over CML on one expert: the additional nonlinearity does not help performance. The next two rows use approximations in Gibbs sampling. When each expert is trained with CML, this replaces the sampling from the expert parameters by finding the maximum posterior estimate given the other model parameters discussed in section 5.3.2. Though this is not exact, it is easier to implement. The results are similar. The bottom row shows the result of replacing Gibbs sampling with large-margin training for the expert parameters discussed in section 5.3.3. Training the mixture of experts this way compared to training a single expert yields a 0.23% absolute improvement in word error rate (WER).

As shown in Table 6.5, compared with the LLMs (with "CML" and "Large Margin" criteria), the overall performance gains for the iLLMs (with "CML*" and "Large Margin*" criteria) are limited, from 8.05% and 7.86% to 8.04% and 7.63%. The iLLM is a mixture-of-expert model, which employs multiple experts to yield a non-linear decision boundary in the classification task. For the iLLM, there could be unbounded number of experts, and a complicated decision boundary can be produced. Although the iLLM can take advantage of the non-linearity, the performance gains are still limited. One possible reason is the limitation of the gating network, which gives the probabilities of observations associated with different experts. In this work, the gating network is based on the iGMM (as described in sections 4.3.5 and 5.3). Figure 6.3 illustrates the change of the number of the represented experts in training. As shown in the figure, after around 400 iterations, the number of experts

	one	two	three	four	five	six	seven	eight	nine	zero	oh	Del
one	5516	12	36	131	18	4	21	2	113	15	19	218
two	1	5510	67	15	2	54	8	29	10	55	35	229
three	2	167	5603	13	6	14	7	31	12	13	12	120
four	41	29	13	5581	8	19	6	6	7	21	80	99
five	8	5	11	11	5802	7	18	2	114	4	81	47
six	3	72	41	22	7	5541	29	40	4	20	15	141
seven	11	16	14	11	4	44	5728	3	14	28	25	62
eight	9	44	84	22	9	97	12	5136	33	14	86	329
nine	21	8	20	5	38	3	12	7	5656	17	21	107
zero	5	79	17	10	1	11	8	7	18	5739	26	99
oh	1	10	17	109	30	2	18	19	26	42	5455	221
Ins	4	85	37	27	15	121	11	88	25	21	103	–

Table 6.6: *The confusion matrix corresponds to the results of the iLLM (Large Margin*) on the test set A of the AURORA 2 corpus.*

becomes stable. The gating network might not be able to fully explore the space. Different forms of the gating network can be studied as a future work.

The classification errors can also be presented in the form of a confusion matrix, which characterises how often a presented word (or phone) was recognised or confused with response alternatives [127]. Table 6.6 shows the confusion matrix corresponding to the results of the iLLM (Large Margin*) (given in Table 6.5) on the test set A of the AURORA 2 corpus. The matrix element C_{ij} denotes how often the word in row i is classified as the word in column j. Thus, the diagonal elements show the number of correctly classified words, and the off-diagonal elements show the number of misclassifications, i.e. substitution (Sub), deletion (Del) and insertion (Ins) errors [162, 186]. As shown in Table 6.6, the iLLM tends to have high deletion errors, which are given in the last column. Thus, the high deletion errors limit the performance of the iLLM.

6.2 Experiments on AURORA 4

In the previous section, the experiments conducted on the AURORA 2 corpus were discussed. Since AURORA 2 is a small vocabulary noise corrupted continuous digit database, the unstructured models can be applied. Each utterance is segmented into words which are treated independently, then each isolated word can be recognised by using the unstructured

models. The performance of the structured models were also examined in the previous section, and the structured models outperform the unstructured models. Since AURORA 4 is a medium vocabulary continuous speech recognition task, isolated word recognition becomes impractical. Thus, only the results of structured discriminative models will be discussed in this section.

6.2.1 The AURORA 4 Corpus

AURORA 4 is a noise-corrupted medium vocabulary corpus. There are two types of training sets, namely the clean training set and the multi-condition training set. The clean set is the standard SI-84 WSJ0 set, which consists of 7138 utterances from 83 speakers and has 14 hours of speech. The multi-condition training set is artificially corrupted from the clean training set with different noise and channel conditions. The test set is based on the development set of 1992 November NIST evaluation, and it is artificially corrupted by using 6 types of noise under 2 channel conditions. The test set consists of 4 sets: A, B, C and D. Set A is clean, set B has 6 types of additive noise, set C has channel distortion, and set D has both additive noise and channel distortion.

6.2.2 Experiments Setup

For the VTS compensated HMM system, the 39-dimensional features are used, which consist of 12 MFCCs, appended with zeroth cepstrum, delta and delta delta coefficients. The HMMs are initially trained with the clean training data, and then compensated with VTS compensation [2]. The cross-word context-dependent triphone models are used, and each model has 3 emitting states. Decision tree based state clustering [202] is applied to define 3143 states, and each one has 16 Gaussian components.

In order to examine the performance of the log-linear models based on different systems, the tandem and hybrid systems are also used, and they are trained with the multi-condition training data. For the tandem system, context-dependent triphone HMMs with 3 emitting states are used, and there are 3033 distinct states (defined by decision tree based state clustering) with an average of 8 Gaussian components per state. The input features are 65 dimensions, and the features consists of the 26 dimensional MLP features and the 39

System	Criterion	Test Set WER(%)				Avg.
		A	B	C	D	
VTS-HMM	ML	7.05	15.21	11.89	23.03	17.74
LLM	CML	7.16	14.86	11.39	22.78	17.46
	Large Margin	7.55	14.22	11.31	21.89	16.83
Tandem	MPE	7.15	11.06	14.37	24.54	16.79
LLM	CML	6.95	11.00	14.29	24.39	16.68
	Large Margin	7.02	10.92	14.16	24.28	16.60
Hybrid	MPE	3.96	7.64	7.79	18.51	12.05
LLM	CML	3.94	7.53	7.36	18.38	11.91
	Large Margin	3.66	7.59	7.47	17.99	11.76

Table 6.7: *The performance of the LLMs based on the VTS-HMM, tandem and hybrid systems on the AURORA 4 corpus*

dimensional PLP features (including zeroth cepstrum, delta, delta-delta and triples coefficients followed by a HLDA projection [116]). For the hybrid system, there are 3033 context-dependent distinct states, and the features are 72-dimensional FBK+Δ+Δ^2 features[1]. The structure of the deep neural network (DNN) is $792 \times 2000^5 \times 3033$, and 11 consecutive frames are concatenated as the input of the DNN.

6.2.3 The Structured Discriminative Models

In experiments, the joint features for the structured discriminative models are based on the log-likelihood features. As discussed in section 3.5, the form of the joint feature vector can be expressed as:

$$\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) = \begin{bmatrix} \sum_{i=1}^{I} \delta(w_i, v_1)\varphi(\boldsymbol{O}_{(i)}) \\ \vdots \\ \sum_{i=1}^{I} \delta(w_i, v_L)\varphi(\boldsymbol{O}_{(i)}) \\ \log\big(P(W)\big) \end{bmatrix} \tag{6.15}$$

where $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \ldots, \boldsymbol{O}_{(I)}\}$ is an utterance, $W = \{w_1, \ldots, w_I\}$ is the sentence label and $\boldsymbol{\rho}$ is the corresponding segmentation. $P(W)$ is the probability given by the language mod-

[1] The FBK stands for the filter bank. The FBK features have 24 dimensions; Δ and Δ^2 denote the delta and delta-delta coefficients.

System	Criterion	Test Set WER(%)				Avg.
		A	B	C	D	
Tandem	MPE	4.78	7.63	8.93	19.14	12.45
Hybrid		3.75	6.70	7.68	17.62	11.24
Joint	Manual	3.79	6.47	7.86	17.34	11.04
LLM	Manual	3.74	6.57	7.88	17.12	10.98
	CML	3.66	6.56	7.88	17.06	10.94
	Large Margin	3.64	6.56	7.04	16.83	10.79

Table 6.8: *The performance of the log-linear models based on the joint decoding system on the AURORA 4 corpus*

els. In this work, the tri-phone models for the generative models are tied to be mono-phones for the discriminative models. Thus, in the joint feature definition (6.15), $\{v_1, \ldots, v_L\}$ are all the mono-phones in the vocabulary, and the log-likelihood features $\varphi\left(\boldsymbol{O}_{(i)}\right)$ are based on the tri-phone models as described in equation (3.70). In experiments the VTS-HMM, tandem and hybrid systems were used in the feature generation.

In training the log-linear models (LLM), the conditional maximum likelihood (CML) estimation (described in section 3.3.1) and large margin training (in section 3.2.5.1) were used. Table 6.7 gives the performance of different LLMs based on the VTS compensated HMM, tandem and hybrid systems. As shown in this table, training the log-linear model with conditional maximum likelihood results in a 0.1-0.3% absolute improvement. Using a large-margin criterion instead consistently improves performance further. 0.9% absolute for the VTS-HMM system, and around 0.1% absolute for the tandem and hybrid systems. Most notably, on the sequence-trained hybrid system the log-linear model improves from 12.05 to 11.76%.

6.2.3.1 *Based on Multiple Systems*

In the previous subsection, the log-likelihood features are based on the likelihoods generated by the VTS-HMM, tandem or hybrid system. As discussed in section 3.5.2.2, the log-likelihood features can also be based on multiple systems. In this section, the features generated by both the tandem and hybrid systems will be examined. Then, in the joint

System	Test Set WER(%)				Avg.
	A	B	C	D	
VAT [193]	5.60	11.00	8.80	17.80	13.40
DNN [40]	5.60	8.80	8.90	20.00	13.40
DNN + dropout [159]	5.40	8.30	7.60	18.50	12.40
Joint (this thesis)	3.79	6.47	7.86	17.34	11.04

Table 6.9: *Comparison of different systems in the literature on the AURORA 4 corpus*

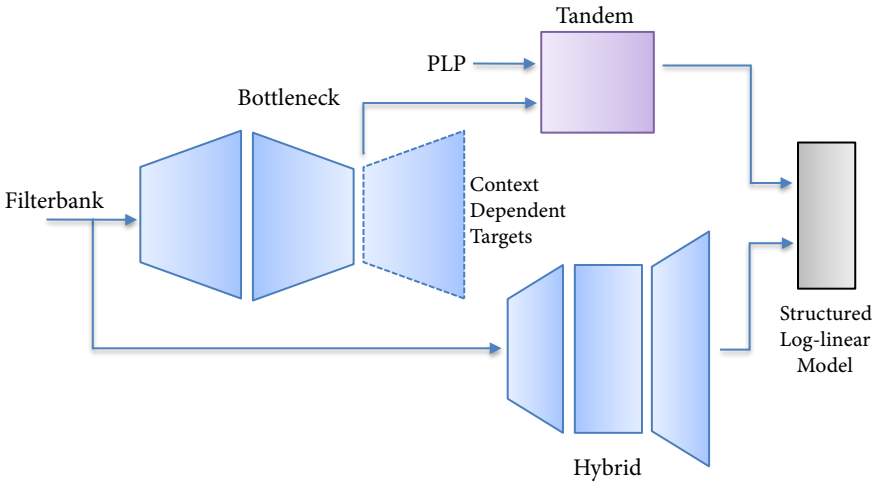


Figure 6.4: *The system framework used for the AURORA 4 corpus.*

feature vector described in (6.15), the log-likelihood features $\varphi(\boldsymbol{O}_{(i)})$ can be expressed as:

$$\varphi(\boldsymbol{O}_{(i)}) = \begin{bmatrix} \log p_{\text{tandem}}(\boldsymbol{O}_{(i)}|w_i) \\ \log p_{\text{hybrid}}(\boldsymbol{O}_{(i)}|w_i) \end{bmatrix} \tag{6.16}$$

where w_i is the corresponding label of $\boldsymbol{O}_{(i)}$. In this work, only the log-likelihoods associated with the correct label were used in the features based on multiple systems, and two systems were used in the feature generation, i.e. the tandem and hybrid systems.

The experimental results of the LLMs using the features based on the tandem and hybrid systems were tabulated in Table 6.8. Compared with the tandem and hybrid systems in Table 6.7, better baseline systems were used for the LLMs based on multiple systems as shown in Table 6.8, i.e. 12.25% and 11.24% WER for the tandem and hybrid systems respectively. The performance gains are mainly due to the use of 40 dimensional FBK rather than 24 dimensional FBK features for the hybrid system, and the use of bottle neck features based on the FBK rather than on the PLP features for the tandem system [199]. The system frame-

System	Criterion	Test Set WER(%)				Avg.
		A	B	C	D	
VTS-HMM	ML	7.05	15.21	11.89	23.03	17.74
LLM	CML	7.16	14.86	11.39	22.78	17.46
	Large Margin	7.55	14.22	11.31	21.89	16.83
iLLM	Bayesian	7.2	14.85	11.42	22.72	17.43
	CML*	7.16	14.85	11.4	22.79	17.46
	Large Margin*	7.55	14.17	11.39	21.81	16.77

Table 6.10: *The performance of the iLLMs on the AURORA 4 corpus. "Bayesian" denotes Bayesian inference of the iLLM, where Gibbs sampling is used. "$*$" indicates approximations are made in Gibbs sampling. "CML*" means sampling for each expert is replaced by CML training, and "Large Margin*" denotes sampling for each expert is replaced by large margin training.*

work is illustrated in Figure 6.4. The baseline joint decoding system used in this thesis has the state-of-the-art performance. A comparison between the published results is given in Table 6.9.

In Table 6.8, "Joint" denotes the joint decoding system introduced in section 2.2, where the log-likelihoods are combined at frame level as described in equation (2.10). In experiments, the system combination weights were manually set to be 0.2 and 1.0 (corresponding to the tandem and hybrid systems respectively). By using joint decoding, 0.2% performance gains can be achieved compared with the baseline hybrid system. For the LLM, the segment level log-likelihoods were used. In the manual setting the weights corresponding to the tandem and hybrid systems were set to be 0.2 and 1.0 respectively. The LLM (with the manual setting) can achieve around 0.1% WER reduction compared with joint decoding. By using CML estimation, the WER can be further reduced, i.e. 10.94% on average. Most notably, by using large margin training, the WER can be reduced from 11.04% to 10.79%.

6.2.4 *The Infinite Structured Discriminative Models*

In this subsection, the experimental results of the infinite structured discriminative models will be discussed. Same as those on the AURORA 2 corpus, Bayesian inference of the infinite log-linear model (iLLM) discussed in section 5.3.1 and two approximate approaches will be

Language	ID	Release
Swahili	202	IARPA-babel202b-v1.0d
Tok Pisin	207	IARPA-babel207b-v1.0b
Lithuanian	304	IARPA-babel304b-v1.0b
Javanese	402	IARPA-babel402b-v1.0b

Table 6.11: *The Babel languages used in this thesis.*

examined. For these approximate methods, sampling of the each expert's parameters were replaced by CML estimation or large margin training, as described in sections 5.3.2 and 5.3.3. The experimental results are given in Table 6.10, in which the performance of the baseline parametric models are also given. This table shows the same set of experiments carried on the AURORA 2 corpus. The results are similar. There is no discernible difference between methods that train a probabilistic criterion, whether Bayesian or CML. However, the large margin criterion shows a good improvement, and optimising a large margin criterion inside an infinite mixture of experts yields a small additional improvement.

6.3 Experiments on Babel

In the previous sections, the experiments conduced on the AURORA 2 and AURORA 4 corpora have been discussed. For these two corpora, the noises are artificially added, and the language used is English. In this section the experiments carried on the Babel corpora will be discussed. In this work, two types of Babel data were used, i.e. the very limited language pack (VLLP) and the full language pack (FLP), which contain up to 3 hours and 60 hours of transcribed conversational telephone speech data respectively. Four different languages were used in experiments, i.e. Swahili, Tok Pisin, Lithuanian and Javanese. The official releases of these languages are given in Table 6.11. All the baseline systems (including hybrid and tandem systems) are built by the Cambridge speech group in the Babel program. These baseline systems were used in Babel evaluation, and have the state-of-the-art performance [36, 124].

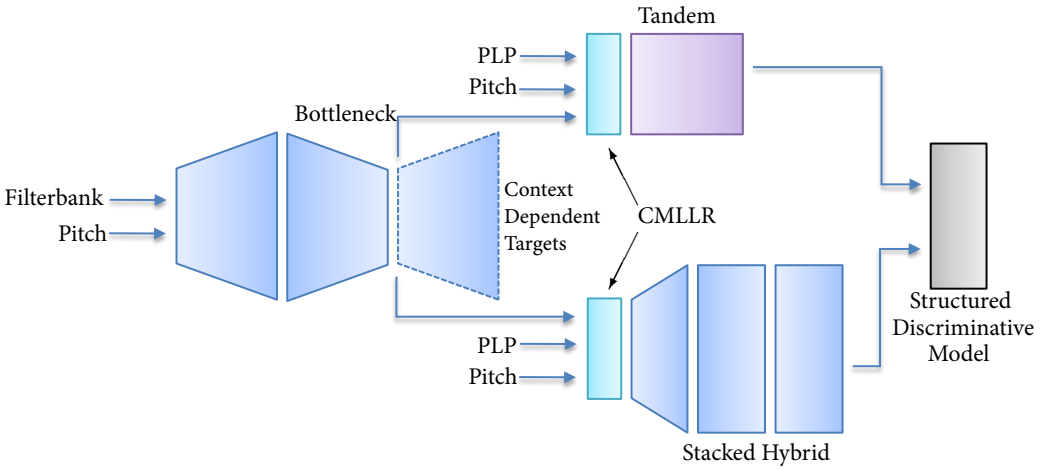


Figure 6.5: *The system framework used for the Babel corpora.*

6.3.1 Experiments Setup

According to the rules of the Babel program, no phonetic lexica may be used. Therefore the systems used graphemic lexica generated using an approach which is applicable to all Unicode characters [59]. The language models were estimated only on the transcripts of the training data. The front-end is an MRASTA based neural network [182], which is initially trained with the data from 11 Babel full language packs, generating 62 dimensional bottleneck features. The input features contain the bottleneck features, 13 PLP coefficients with dynamics of orders 1, 2, and 3, and pitch and probability-of-voicing features (estimated with the Kaldi toolkit [142]) with dynamic coefficients of orders 1 and 2. Two sets of acoustic models are constructed. One is a speaker independent (SI) model, which is based on the tandem features. The other is estimated using speaker adaptive training (SAT) [5]. SAT is performed using global constrained maximum-likelihood linear regression (CMLLR) [55] on the maximum-likelihood trained models, followed by MPE. During training, the supervision for CMLLR is the reference. During testing, the SI model with a tri-gram language model is used to produce hypotheses. The resulting CMLLR transforms are used to obtain speaker-normalised features, which are then input to the Tandem-SAT model. The number of context-dependent states is 1000; each state has an average of 16 components.

As illustrated in Figure 6.5, the stacked hybrid system use the same features as the tandem system, derived from the CMLLR transforms generated by the tandem SAT system. The input to the hybrid DNN is a concatenation of 9 consecutive feature vectors. The network has layer sizes of $963 \times 1000^4 \times 1000$ and is initialised by layer-wise pre-training with

Language	System	Criterion	dev set TER (%)
Swahili	Hybrid SI	MPE	61.3
	LLM	Large Margin	60.7
	Hybrid SAT	MPE	60.5
	LLM	Large Margin	59.9
Tok Pisin	Hybrid SAT	MPE	52.7
	LLM	Large Margin	52.5
Lithuanian	Hybrid SAT	MPE	63.2
	LLM	Large Margin	62.9

Table 6.12: *The performance of the log-linear model on different Babel VLLP corpora*

context-independent targets. Finetuning is done using the frame-level cross-entropy criterion with context-dependent targets. The number of context-dependent states is the same as in the tandem system. Then, sequence training using the MPE criterion is applied for further improvement. As a comparison, a joint decoding system [191], which applies log-linear combination of the tandem and hybrid systems at the frame level, is used. The frame-level log-likelihoods from the tandem system are multiplied by 0.25, and those from the hybrid system by 1. They are then added at the frame level and used instead of the normal HMM log-likelihoods, as described in (2.10).

6.3.2 *The Structured Discriminative Models*

In experiments, similar to the LLMs used on the AURORA 4 corpus, features based on log-likelihoods were used. For the experiments on the Babel corpora, graphemes are used rather than phones. The joint features have the same format as those discussed in section 3.5.2, which also applies to graphemes. Same as the experiments on AURORA 4, the graphemes with context information are used for the generative models, and they are tied to be the central graphemes for the discriminative models, namely for the joint features as described in (6.15), central graphemes v_l are used, but the likelihoods are based on the graphemes with context information.

Table 6.12 gives the results of LLMs on different languages of Babel VLLP corpora. The first block (4 rows) examine the effect of speaker-dependent transformations of the acoustic features for a hybrid system on the performance of the LLM. The first two rows give results

System	Criterion	Test Set TER(%)	
		dev set	eval set
Tandem SAT	MPE	62.5	63.0
Hybrid SAT		60.5	59.4
Joint Decoder	Manual	59.4	58.3
LLM	Manual	58.9	57.9
	Large Margin	57.9	56.8

Table 6.13: *The results on the Babel_202 Swahili VLLP corpus. For the manual setting, the weights corresponding to the hybrid log-likelihoods are set to 1, and the weights corresponding to the tandem log-likelihoods are set to 0.25.*

of the speaker-independent system. The LLM improves performance by 0.6% absolute. The third and fourth lines have the same contrast, but based on a speaker-dependent HMM. The increase of performance from the LLM is also 0.6%. Speaker-dependent transformations therefore do not appear to decrease the usefulness of the features derived from the HMM. The rest of the table examines how performance improvement varies for different languages. For Tok Pisin, the improvement is 0.2, and for Lithuanian 0.3. Though the performance increase does vary with languages, there is consistently an increase.

Results of experiments with combining tandem and hybrid systems on Babel_202 Swahili VLLP are in Table 6.13. The top block repeats the tandem and hybrid HMM baselines. The next block shows the performance of the joint decoding system [191], which performs frame-level combination. The weights are fixed to 0.25 for the tandem system and 1 for the hybrid. The next line shows the result of the LLM that uses the same manual configuration (0.25 and 1 for the tandem and hybrid systems respectively). On the development set, performance improves by 0.5% compared to the joint system, probably caused by the difference in the assignment of the underlying HMM states to time frames. Firstly, the likelihoods are used, so the sum over all paths instead of the one best path is used, and secondly, those paths can be different between the tandem and hybrid systems, allowing a more optimal alignment for both. When the parameters of the log-linear model are trained, for the bottom line of Table 5, performance increases further by 1%. The absolute improvement over the joint system is 1.5%. When applying the LLMs with the same configurations to the evaluation set, the same trend can be observed. The LLM with manual setting outperforms joint decoding,

System	Criterion	dev set TER(%)
Tandem SAT	MPE	60.8
Hybrid SAT		60.0
Joint Decoder	Manual	58.6
LLM	Manual	58.5
	Large Margin	57.7

Table 6.14: *The results on the Babel_402 Javanese FLP corpus (46 hours). For the manual setting, the weights corresponding to the hybrid log-likelihoods are set to 1, and the weights corresponding to the tandem log-likelihoods are set to 0.25.*

and 0.4% performance gain can be achieved. When the LLM are trained with large margin criterion, further gain can be obtained, with 1.1% absolute token error rate (TER) reduction.

Same set of experiments were also carried on the Babel_402 Javanese FLP corpus, which contains 46 hours' speech data. As tabulated in Tabel 6.14, the performance of the LLM is consistent with that on the Babel_202 Swahili VLLP corpus. The LLM with the manual setting slightly outperforms joint decoding, with 0.1% TER reduction. By applying large margin training to the LLM, the TER can be further decreased by 0.8%.

In order to ensure efficient training, in this work, all training data (numerator and denominator lattices) are loaded in the memory. When applying FLP data to LLMs, this might be a problem. As shown in Table 6.15, if all data are loaded, the memory use is 230 GB. By using a subset of the training data, training would be more practical. Table 6.15 gives the results of the LLMs with various amount of training data. These subset are randomly sampled from the whole training data, and equal amount of data are sampled for each speaker. The second block of Table 6.15 gives the performance of the LLM trained with various amount of data. As the increase of the amount of training data, the token error rate (TER) reduces. When the data amount increases to around 2 hours (with memory use around 11 GB), the TER cannot be further lowered. Training with more data, the performance of the LLM stays the same (with TER 57.7%). Thus, when the amount of training data is large, a subset can be used in training. As shown in this table, with relatively small amount of data, the LLM also yield performance gains. This motives the application of LLMs to adaptation, which would be one option for the future work.

System	Criterion	Data Hours	Mem Use (GB)	Test Set TER(%)
Joint Decoder	Manual	46.8	—	58.6
LLM	Large Margin	46.8	230.0	57.7
LLM	Large Margin	4.52	22.69	57.7
		2.26	11.23	57.7
		1.49	8.14	58.2
		1.13	5.6	58.2
		0.68	3.5	58.5
		0.23	1.3	58.5

Table 6.15: *The performance of the log-linear with various hours of training data on the BABEL_402 Javanese FLP corpus. For the manual setting, the weights corresponding to the hybrid log-likelihoods are set to 1, and the weights corresponding to the tandem log-likelihoods are set to 0.25.*

Chapter 7

Conclusions

This thesis investigated structured discriminative models for speech recognition. The previous work in this area has been extended in two directions. First is the study of the features generated by a single system and multiple systems based on deep neural networks (DNNs), the form of these features were discussed in section 3.5. In experiments, LLMs using these features were examined on various type of datasets with different training criteria, and consistent performance gains can be achieved. The second extension is the use of the mixture-of-experts framework. By using this framework, LLMs can be extended to infinite LLMs, where different experts are employed to focus on different regions of the feature space to make an ensemble decision, and an non-linear overall decision boundary can be yielded. This thesis also discussed a criterion based perspective on Bayesian inference, where Bayesian inference can be viewed as a minimisation criterion, which consists of two terms that represent prior beliefs and information from the observations as discussed in section 5.2. By using this criterion, different criterion function (which represents information from the observations) can be used, and various meaningful training criteria can be resulted, such as large margin training. As discussed in sections 5.3 and 5.4, these criteria can be applied to the infinite LLM. In experiments Bayesian inference of the infinite LLM and its variants (discussed in section 5.3) were examined. Bayesian inference of the iLLM does not yield significant performance gains. When each expert is trained with large margin criterion, small gains can be achieved.

7.1 Future Work

These are many possible directions for the future work, and a number of suggestions are given as follows:

- In this thesis, the features used by the LLMs are at most based on two systems, i.e. the tandem and hybrid systems. Using features from two systems might be too limited. In the future work, features based on more than two systems can be used.
- As discussed in section 3.5, in general the features for discriminative models can be categorised into two groups, i.e. the frame level and segment level features. In experiments, only the segmental level features were examined, and the study of the frame level features can be one possible direction for the future work.
- When applying Large margin training to LLMs, which are equivalent to structured SVMs, performance gains can be achieved with a small amount of training data, as shown in Table 6.15. This motivates the application of LLMs (with large margin training) to adaptation, where only a small amount of data for target speaker is available.
- In this work, tri-phone models are used for the generative models (HMMs), and the tri-phones are tied to be mono-phones for the discriminative models. The more general parameters tying approach based on decision trees [148] can be used as a future work.
- In the past few years, models with deep architecture, such as deep neural networks (DNNs), have been extensively used in speech community, and significant performance gains have been achieved compared with the models having shallow architecture. In this thesis, although the features extracted from the DNNs are used by the structured discriminative models, the models themselves only have shallow architecture. By introducing deep architecture into the structured models, the modelling capacity can be improved significantly.
- For the infinite LLMs, in experiments only Bayesian inference and its two variants (approximate approaches) discussed in section 5.3 were examined. In these two approximate approaches, sampling for each expert's parameters are replaced by CML estimation and large margin training, and large margin training can yield the best results for the infinite LLMs. This motives the application of large margin training

to the whole infinite LLM (the gating network and the experts), and the training approach was discussed in section 5.4.

- For the infinite LLMs discussed in Chapter 5, the indicator variable corresponding to each utterance is a scalar, where the inputs to gating network are utterances and all the segments in an utterance share the same indicator. This might limit the flexibility of the gating network. In order to make better use of the data, a more granular (vector) indicator can be introduced. By doing so, a mixture of experts for each sub-sentence unit (such as word or phone) could be built, and this type of model is called the structured infinite discriminative model, which is discussed in Appendix F.

Appendices

Appendix A

Probability Measures

In mathematics, a *measure* $\boldsymbol{G}$ is a function that assigns a non-negative real number to subsets of a set $\boldsymbol{\Theta}$, which is sometimes called the *sample space*. It must assign 0 to the empty set $\varnothing$, and be countably additive. A measure is a generalisation of the concepts of length, area, and volume. If measure $\boldsymbol{G}$ assigns 1 to the entire measurable set $\boldsymbol{\Theta}$, measure $\boldsymbol{G}$ is a *probability measure* (or *probability distribution*). Formally, the probability measure can be defined as follows [87, 184]:

A probability measure is a non-negative function $\boldsymbol{G}$ defined for countable collections of mutually disjoint sets $\{\boldsymbol{A}_n\}_{n=1}^{N}$ with $\boldsymbol{A}_n \in \Sigma_{\boldsymbol{\Theta}}$ and $\boldsymbol{A} = \cup_n \boldsymbol{A}_n$ that satisfies the following properties:

$$\boldsymbol{G}(\boldsymbol{A}_n) \geq 0 \tag{A.1}$$

$$\boldsymbol{G}(\boldsymbol{\Theta}) = 1 \quad \text{and} \quad \boldsymbol{G}(\varnothing) = 0 \tag{A.2}$$

$$\boldsymbol{G}(\boldsymbol{A}) = \sum_n \boldsymbol{G}(\boldsymbol{A}_n) \tag{A.3}$$

$$\boldsymbol{G}(\boldsymbol{A}_n^{\mathsf{c}}) = 1 - \boldsymbol{G}(\boldsymbol{A}_n) \tag{A.4}$$

where $\boldsymbol{A}_n^{\mathsf{c}}$ is the complement of $\boldsymbol{A}_n$, namely $\boldsymbol{A}_n^{\mathsf{c}} = \boldsymbol{\Theta} \setminus \boldsymbol{A}_n$. $\Sigma_{\boldsymbol{\Theta}}$ is a *σ-algebra* on set $\boldsymbol{\Theta}$, which is a collection of subsets of $\boldsymbol{\Theta}$ that is closed under countably many set operations (complement, union of countably many sets and intersection of countably many sets). The members of $\Sigma_{\boldsymbol{\Theta}}$ are called *measurable sets*, the pair $(\boldsymbol{\Theta}, \Sigma_{\boldsymbol{\Theta}})$ is called a *measurable space*,

and the triple $(\boldsymbol{\Theta}, \Sigma_{\boldsymbol{\Theta}}, \boldsymbol{G})$ is called a *probability space*. More generally, the triple is called a *measure space* without requiring the measure $\boldsymbol{G}$ to be a probability measure.

Appendix B

Infinite Support Vector Machines

Section 4.3.5 introduced the framework of the infinite mixture of experts with a gating network based on the infinite mixture model. The *infinite support vector machine* (iSVM) introduced by Zhu [213] is a specification of this framework, where each expert (or the zth) is a discriminant function:

$$F(w, \boldsymbol{x}; \boldsymbol{H}, z) = F(w, \boldsymbol{x}; \boldsymbol{\eta}_z) = \boldsymbol{\eta}_z^{\mathsf{T}} \phi(\boldsymbol{x}, w) \tag{B.1}$$

where $\boldsymbol{x}$ is the input variable, and w is the corresponding class label, which takes value from the set $\{1, \ldots, L\}$. $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$ are the parameters of the the experts. z is the indicator variable that denotes which expert the input $\boldsymbol{x}$ is associated with. The feature function $\phi(\boldsymbol{x}, w)$ is defined as:

$$\phi(\boldsymbol{x}, w) = \begin{bmatrix} \delta(w, 1)\boldsymbol{x} \\ \vdots \\ \delta(w, L)\boldsymbol{x} \end{bmatrix} \tag{B.2}$$

where $\delta(\cdot)$ is a Kronecker delta. In a discriminant function, given an input, the class label can be obtained by maximising this discriminant function. In (B.1), a discriminant function for the zth expert is described. The overall discriminant function for the iSVM can be described as a summation over all experts:

$$F(w, \boldsymbol{x}) = \sum_{z} \int F(w, \boldsymbol{x}; \boldsymbol{H}, z) \hat{q}(\boldsymbol{H}, z) d\boldsymbol{H} \quad z \in \{1, 2, \cdots, \infty\} \tag{B.3}$$

where $\hat{q}(\boldsymbol{H}, z)$ is an optimal distribution obtained in training, which will be discussed in detail in the following sections.

B.1 The Training Criterion

The iSVM is a specification of the infinite mixture of experts discussed in section 4.3.5. An mixture of experts consists of two parts, namely the gating network and experts. In this section, training criteria for the gating network and experts will be introduced first, and an overall criterion for the iSVM will be discussed subsequently.

B.1.1 The Training Criterion for the Experts

In training the iSVM, maximum entropy discrimination (MED) [90, 93] is applied. MED is a large margin training approach, through which an optimal distribution of the model parameters (such as $\hat{q}(\boldsymbol{\eta})$) can be estimated rather than optimal values (such as $\hat{\boldsymbol{\eta}}$). Take a single classifier with discriminant function $F(w, \boldsymbol{x}; \boldsymbol{\eta})$ for example, given the training data $\mathcal{D} = \{(\boldsymbol{x}_1, w_1), \cdots, (\boldsymbol{x}_N, w_N)\}$, the MED training criterion can be described as:

$$\begin{aligned} \arg\min_{q(\boldsymbol{\eta})} \Bigg\{ \mathrm{KL}\big(q(\boldsymbol{\eta})||p(\boldsymbol{\eta})\big) + C \sum_n \Bigg[\max_{w \neq w_n} \Big\{ \mathcal{L}(w, w_n) + \\ \int q(\boldsymbol{\eta}) \Big(F(w, \boldsymbol{x}_n; \boldsymbol{\eta}) - F(w_n, \boldsymbol{x}_n; \boldsymbol{\eta}) \Big) d\boldsymbol{\eta} \Big\} \Bigg]_+ \Bigg\} \\ \text{s.t.} \quad q(\boldsymbol{\eta}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{B.4}$$

where the best competing hypothesis w is found for all possible labels except the correct one w_n. $\mathrm{KL}\big(q(\boldsymbol{\eta})||p(\boldsymbol{\eta})\big)$ is the Kullback-Leibler (KL) divergence [1], and $p(\boldsymbol{\eta})$ is the prior distribution. $[\cdot]_+$ is the hinge loss which is defined in (5.10). The loss $\mathcal{L}(w, w_n)$ measures how different between the labels w and w_n. $\mathcal{P}_{\text{prob}}$ is the set consisting of all possible valid distribution over $\boldsymbol{\eta}$. The discriminant function $F(w, \boldsymbol{x}; \boldsymbol{\eta})$ is defined in (B.1). C is a non-negative constant, which is used to balance the regularisation term (the KL divergence) with the training loss (the hinge loss function).

As discussed in section 4.3.5.3, the iSVM is a specification of the infinite mixture of expert. In inference of the iSVM, a representation of the iSVM based on the stick-breaking

[1] The KL divergence is defined as: $\mathrm{KL}\big(q(\boldsymbol{\eta})||p(\boldsymbol{\eta})\big) = \int q(\boldsymbol{\eta}) \log\big(\frac{q(\boldsymbol{\eta})}{p(\boldsymbol{\eta})}\big) d\boldsymbol{\eta}$.

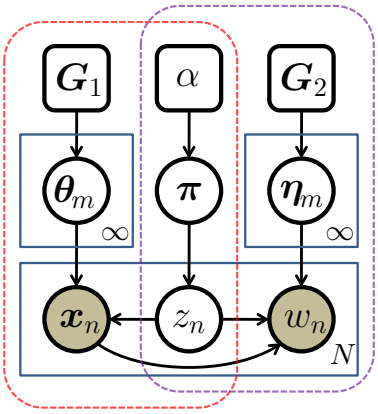


Figure B.1: *The graphical model of the iSVM. The plot associated with the red dotted line is the stick-breaking construction of the infinite mixture model.*

process is used, and this representation is illustrated in Figure B.1, which is the same as the graphical model of the infinite mixture of expert described in Figure 4.9. Analogous to the generative process of the infinite mixture of expert described in (4.48), in the iSVM, the process of determining which expert is used to classify an observation can be described as follows [213]:

$$v_m \sim \text{Beta}(1, \alpha) \tag{B.5}$$

$$\pi_m = v_m \prod_{i=1}^{m-1} (1 - v_i) \tag{B.6}$$

$$\boldsymbol{\eta}_m \sim \boldsymbol{G}_2 \tag{B.7}$$

$$z_n \sim \text{Categorical}(\boldsymbol{\pi}) \tag{B.8}$$

where the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are given by the stick-breaking process (4.26), which is normally denoted as $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ [138, 160]. $\boldsymbol{G}_2$ is the prior distribution. z_n is the indicator variable that denotes with which expert the nth observation is associated. $\text{Beta}(\cdot)$ is the beta distribution. $\text{Categorical}(\cdot)$ is the categorical distribution, which is the generalisation of the Bernoulli distribution with multiple possible outcomes.

When applying the MED training criterion described in (B.4) to the iSVM which has infinite number of experts, in order to specify which expert the input is associated with, indicator variables $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ corresponding to the training data are introduced. Then, for the iSVM, the hinge loss function in the MED criterion can be described as fol-

lows [213]:

$$\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big) = \sum_{n=1}^{N}\Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}(w,w_n)+ \int\sum_{z_n} q(\boldsymbol{H},z_n)\Big(F(w,\boldsymbol{x}_n;\boldsymbol{H},z_n)-F(w_n,\boldsymbol{x}_n;\boldsymbol{H},z_n)\Big)d\boldsymbol{H}\Big\}\Bigg]_+ \tag{B.9}$$

Analogous to the MED criterion (B.4) for a single classifier, the MED criterion for the iSVM can be described as follows [213]:

$$\begin{aligned}&\arg\min_{q(\boldsymbol{H},\boldsymbol{z})}\Big\{\mathrm{KL}\big(q(\boldsymbol{H},\boldsymbol{z})||p(\boldsymbol{H},\boldsymbol{z})\big)+C\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)\Big\}\\ &\text{s.t.}\quad q(\boldsymbol{H},\boldsymbol{z})\in\mathcal{P}_{\text{prob}}\end{aligned} \tag{B.10}$$

where the hinge loss function $\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)$ is defined in (B.9).

B.1.2 The Training Criterion for the Gating Network

As illustrated in Figure B.1, in the iSVM the underlying distribution of the observations $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ is also modelled by the gating network, which is a mixture model in the iSVM. According to the graphical model of the gating network, which is the plot associated with the red dotted line in Figure B.1, the generative process corresponding to the gating network can be described as follows:

$$\boldsymbol{\pi}\sim\mathrm{GEM}(\alpha) \tag{B.11}$$

$$\boldsymbol{\theta}_m\sim\boldsymbol{G}_1 \tag{B.12}$$

$$z_n\sim\mathrm{Categorical}(\boldsymbol{\pi}) \tag{B.13}$$

$$\boldsymbol{x}_n\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_n}) \tag{B.14}$$

where the $\boldsymbol{\pi}=\{\pi_m\}_{m=1}^{\infty}$ are generated in the stick-breaking process described in (B.5) and (B.6). The component likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ is given by a broad class of distributions called the *exponential family* [11, 43] having the following form:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x})\exp\big(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}-A(\boldsymbol{\theta})\big) \tag{B.15}$$

where $\boldsymbol{\theta}$ is called the *natural parameter* of the distribution. $h(\boldsymbol{x})$ and $A(\boldsymbol{\theta})$ are known functions, and $A(\boldsymbol{\theta})$ can be interpreted as the coefficient that ensures that the distribution

is normalised [16]. In Bayesian inference, the posterior distribution of the model parameters for the gating network is $p(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z}|\mathcal{D})$[1]. Thus, in variational inference the (approximate) distribution to be inferred for the gating network is:

$$\begin{aligned} &\arg\min_{q(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{z})} \mathrm{KL}\big(q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z})||p(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z}|\mathcal{D})\big) \\ &\quad \text{s.t.} \quad q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{B.16}$$

where $\boldsymbol{v} = \{v_m\}_{m=1}^{\infty}$ are the weights in the stick-breaking process as described in (B.5), which are beta distributed.

B.1.3 The Overall Training Criterion for the iSVM

The training criteria for the experts (B.10) and gating network (B.16) were discussed in the previous sections. Given these two criteria, the overall training criterion for the whole iSVM (both the gating network and experts) can be described as the combination of the two criteria [213]:

$$\begin{aligned} &\arg\min_{q(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{H},\boldsymbol{z})} \Big\{\mathrm{KL}\big(q(\boldsymbol{H}, \boldsymbol{z})||p(\boldsymbol{H}, \boldsymbol{z})\big) + C\mathcal{R}\big(q(\boldsymbol{H}, \boldsymbol{z})\big) + C_2\mathrm{KL}\big(q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z})||p(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z}|\mathcal{D})\big)\Big\} \\ &\quad \text{s.t.} \quad q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{B.17}$$

where C_2 is another non-negative constant, which is used to trade off between the minimisation criteria for the gating network and experts. $\mathcal{R}\big(q(\boldsymbol{H}, \boldsymbol{z})\big)$ is the hinge loss function defined in (B.9). By sharing the same indicators, the gating network and the experts are closely coupled. By minimising the overall training criterion (B.17), the optimised model is expected to discover the underlying distribution of data and make predictions well on unseen data [213].

B.2 Optimisation with Coordinate Descent

In order to ensure the overall training criterion (B.17) for the iSVM tractable, the mean-field assumption [98, 188] and truncated stick-breaking representation [19] are used. Thus, the

[1] As described in equation (B.6), the weights $\boldsymbol{v}$ and $\boldsymbol{\pi}$ are mutually convertible.

distribution to be inferred $q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})$ is assumed to be fully factorised as follows:

$$q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \approx \prod_{m=1}^{M-1} q(v_m) \prod_{m=1}^{M} q(\boldsymbol{\theta}_m) \prod_{m=1}^{M} q(\boldsymbol{\eta}_m) \prod_{n=1}^{N} q(z_m) \tag{B.18}$$

where M is the number of mixture weights in the truncated stick-breaking process, where the Mth breaking ratio equals 1, namely $q(v_M) = 1$. This implies that the mixture weight $\pi_m = 0$ for all $m > M$, namely the number of components is limited to M. Thus, the parameters for the components and experts are limited to be finite sets $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ and $\boldsymbol{H} = \{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_M\}$. Again, $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ are the indicator variables corresponding to the training data $\mathcal{D} = \{(\boldsymbol{x}_1, w_1), \cdots, (\boldsymbol{x}_N, w_N)\}$. In the variational distribution (B.18), $q(z_n)$ is the categorical distribution with parameters $\boldsymbol{\varphi}_n = \{\varphi_{n,1}, \ldots, \varphi_{n,M}\}$, which is a generalisation of the Bernoulli distribution with multiple possible outcomes. $q(v_m) = \text{Beta}(c_{m,1}, c_{m,2})$ is the beta distribution. In mean-field variational inference, coordinate descent is performed, in which the parameters of each variational distribution (say the parameters of $q(v_m)$) are updated in turn by iteratively minimising the overall training criterion (B.17). The process of updating each set of variational parameters will be discussed in the following sections.

B.2.1 *Updating $\hat{q}(v_m)$ and $\hat{q}(\boldsymbol{\theta}_m)$*

The optimal distributions $\hat{q}(v_m)$ and $\hat{q}(\boldsymbol{\theta}_m)$ can be obtained by minimising the overall training criterion described in (B.17) given all other variational distributions (such as $q(\boldsymbol{H})$ and $q(\boldsymbol{z})$). Since the first two terms $\text{KL}\big(q(\boldsymbol{H}, \boldsymbol{z})||p(\boldsymbol{H}, \boldsymbol{z})\big)$ and $C\mathcal{R}\big(q(\boldsymbol{H}, \boldsymbol{z})\big)$ are not functions of $q(v_m)$ and $q(\boldsymbol{\theta}_m)$, these two terms can be omitted in updating $\hat{q}(v_m)$ or $\hat{q}(\boldsymbol{\theta}_m)$. Therefore, the optimal distributions $\hat{q}(v_m)$ and $\hat{q}(\boldsymbol{\theta}_m)$ can be obtained by minimising the following simplified criterion:

$$\arg\min_{q(\boldsymbol{v}, \boldsymbol{\Theta})} \text{KL}\big(q(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z})||p(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{z}|\mathcal{D})\big) \tag{B.19}$$

This is the standard training criterion in variational inference of the infinite mixture model [19].

B.2.2 Updating $\hat{q}(\boldsymbol{\eta}_m)$

When optimising $\hat{q}(\boldsymbol{\eta}_m)$, all the other variational distributions (such as $\hat{q}(v_m)$, $\hat{q}(\boldsymbol{\theta}_m)$ and $\hat{q}(\boldsymbol{z})$) are given. Then minimisation of the overall training criterion (B.17) becomes minimising:

$$\arg\min_{q(\boldsymbol{H})} \Big\{ \mathrm{KL}\big(q(\boldsymbol{H})||p(\boldsymbol{H})\big) + C\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big) \Big\} \tag{B.20}$$

In the hinge loss function $\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)$ defined in (B.9), the sum over z_n (where $z_n \in \{1,\ldots,M\}$) is inside the maximisation, which means the best competing hypothesis w is found for all experts. In order to make training more efficient, the hinge loss can be relaxed to an upper bound by moving out the sum over z_n from the maximisation [213]:

$$\begin{aligned}\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big) \leq \sum_{n=1}^{N}\sum_{z_n} q(z_n)\Bigg[\max_{w\neq w_n} \Big\{ \mathcal{L}(w,w_n) + \\ \int q(\boldsymbol{H})\Big(F(w,\boldsymbol{x}_n;\boldsymbol{H},z_n) - F(w_n,\boldsymbol{x}_n;\boldsymbol{H},z_n)\Big)d\boldsymbol{H}\Big\}\Bigg]_+\end{aligned} \tag{B.21}$$

As discussed at the beginning of section B.2, $q(z_n)$ is a categorical distribution with parameters $\boldsymbol{\varphi}_n = \{\varphi_{n,1},\ldots,\varphi_{n,M}\}$ and $q(\boldsymbol{H}) = \prod_{m=1}^{M} q(\boldsymbol{\eta}_m)$. Substituting these definitions and the upper bound of the hinge loss function (B.21) in, the training criterion (B.20) can be further written as:

$$\begin{aligned}\arg\min_{q(\boldsymbol{H})} \Bigg\{ \sum_{m=1}^{M} \mathrm{KL}\big(q(\boldsymbol{\eta}_m)||p(\boldsymbol{\eta}_m)\big) + C\sum_{n=1}^{N}\sum_{m=1}^{M}\varphi_{n,m}\Bigg[\max_{w\neq w_n} \Big\{ \mathcal{L}(w,w_n) + \\ \int q(\boldsymbol{\eta}_m)\Big(F(w,\boldsymbol{x}_n;\boldsymbol{\eta}_m) - F(w_n,\boldsymbol{x}_n;\boldsymbol{\eta}_m)\Big)d\boldsymbol{\eta}_m\Big\}\Bigg]_+\Bigg\}\end{aligned} \tag{B.22}$$

Thus this criterion (B.22) can be written as M minimisation criteria:

$$\arg\min_{q(\boldsymbol{H})} \mathcal{F}\big(q(\boldsymbol{H})\big) = \sum_{m=1}^{M} \arg\min_{q(\boldsymbol{\eta}_m)} \mathcal{F}\big(q(\boldsymbol{\eta}_m)\big) \tag{B.23}$$

where $\mathcal{F}\big(q(\boldsymbol{H})\big)$ is the criterion described in (B.22), and the criterion $\mathcal{F}\big(q(\boldsymbol{\eta}_m)\big)$ for the mth expert can be described as follows:

$$\begin{aligned}\mathcal{F}\big(q(\boldsymbol{\eta}_m)\big) = \mathrm{KL}\big(q(\boldsymbol{\eta}_m)||p(\boldsymbol{\eta}_m)\big) + C\sum_{n=1}^{N}\varphi_{n,m}\Bigg[\max_{w\neq w_n} \Big\{ \mathcal{L}(w,w_n) + \\ \int q(\boldsymbol{\eta}_m)\Big(F(w,\boldsymbol{x}_n;\boldsymbol{\eta}_m) - F(w_n,\boldsymbol{x}_n;\boldsymbol{\eta}_m)\Big)d\boldsymbol{\eta}_m\Big\}\Bigg]_+\end{aligned} \tag{B.24}$$

This is the standard MED criterion as described in (B.4). It is worth noting that $\varphi_{n,m}$ can be merged into the loss function $\mathcal{L}(\cdot)$ and discriminant function $F(\cdot)$. Let $\mathcal{L}'(\cdot) = \varphi_{n,m}\mathcal{L}(\cdot)$ and $F'(\cdot) = \varphi_{n,m}F(\cdot)$, the criterion (B.24) becomes an identical form to the standard MED criterion. As discussed in [93, 213], in MED when the prior distribution $p(\boldsymbol{\eta}_m)$ is a Gaussian distribution, the optimal distribution $\hat{q}(\boldsymbol{\eta}_m)$ is also a Gaussian distribution but with different mean. In the iSVM, the prior distribution over $\boldsymbol{\eta}_m$ is $p(\boldsymbol{\eta}_m) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ (where $\boldsymbol{I}$ is the identity matrix), hence the optimal distribution can be described as $\hat{q}(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{I})$. Substituting the form of $\hat{q}(\boldsymbol{\eta}_m)$ and the definition of the discriminant function (B.1) in, the large margin training criterion (B.24) can be further described as a minimisation criterion to estimate $\boldsymbol{\mu}_m$:

$$\mathcal{F}(\boldsymbol{\mu}_m) = \frac{1}{2}||\boldsymbol{\mu}_m||^2 + C\sum_{n=1}^{N}\varphi_{n,m}\Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}(w, w_n) + \Big(\boldsymbol{\mu}_m^{\mathsf{T}}\phi(\boldsymbol{x}_n, w) - \boldsymbol{\mu}_m^{\mathsf{T}}\phi(\boldsymbol{x}_n, w_n)\Big)\Big\}\Bigg]_+ \tag{B.25}$$

Let $\mathcal{L}'(\cdot) = \varphi_{n,m}\mathcal{L}(\cdot)$ and $\phi'(\cdot) = \varphi_{n,m}\phi(\cdot)$, the large margin training criterion (B.25) can be written in the form of the SVM [31, 33]:

$$\mathcal{F}(\boldsymbol{\mu}_m) = \frac{1}{2}||\boldsymbol{\mu}_m||^2 + C\sum_{n=1}^{N}\Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}'(w, w_n) + \Big(\boldsymbol{\mu}_m^{\mathsf{T}}\phi'(\boldsymbol{x}_n, w) - \boldsymbol{\mu}_m^{\mathsf{T}}\phi'(\boldsymbol{x}_n, w_n)\Big)\Big\}\Bigg]_+ \tag{B.26}$$

This is the training criterion of the SVM. The mean $\hat{\boldsymbol{\mu}}_m$ can be estimated by minimising this criterion, then the optimal distribution $\hat{q}(\boldsymbol{\eta}_m) = \mathcal{N}(\hat{\boldsymbol{\mu}}_m, \boldsymbol{I})$ can be obtained.

B.2.3 Updating $\hat{q}(z_n)$

In [213] optimisation of $q(z_n)$ based on the dual form of the large margin training criterion (B.17) is discussed. In this section an alternative approach, in which $q(z_n)$ is optimised based on the primal form of (B.17), will be studied.

As described in the generative process (B.8), the prior distribution over z_n is a categorical distribution with parameters $\boldsymbol{\pi}$. Again, as described in equation (B.6), the weights $\boldsymbol{v}$ and $\boldsymbol{\pi}$ are mutually convertible. Thus, without $\boldsymbol{v}$ (or $\boldsymbol{\pi}$), the prior distribution $p(z_n)$ is hard to evaluated. This leads to hard evaluation of the term $\mathrm{KL}\big(q(\boldsymbol{H}, \boldsymbol{z})||p(\boldsymbol{H}, \boldsymbol{z})\big)$ in the overall training criterion (B.17). In order to simplify optimisation, this KL divergence is relaxed

to an upper bound $\mathrm{KL}\big(q(\boldsymbol{v},\boldsymbol{H},\boldsymbol{z})||p(\boldsymbol{v},\boldsymbol{H},\boldsymbol{z})\big)$. By using this upper bound, the overall training criterion (B.17) is relaxed to [213]:

$$\begin{aligned}\underset{q(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{H},\boldsymbol{z})}{\arg\min}\ \Big\{&\mathrm{KL}\big(q(\boldsymbol{v},\boldsymbol{H},\boldsymbol{z})||p(\boldsymbol{v},\boldsymbol{H},\boldsymbol{z})\big)+C\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)+\\ &C_2\mathrm{KL}\big(q(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{z})||p(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{z}|\mathcal{D})\big)\Big\}\\ \text{s.t.}\quad & q(\boldsymbol{v},\boldsymbol{\Theta},\boldsymbol{H},\boldsymbol{z})\in\mathcal{P}_{\text{prob}}\end{aligned}\tag{B.27}$$

where $\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)$ is the hinge loss function defined in (B.9). Given all the other variational distributions (such as $\hat{q}(v_m)$, $\hat{q}(\boldsymbol{\theta}_m)$ and $\hat{q}(\boldsymbol{\eta}_m)$), when optimising $q(\boldsymbol{z})$, minimisation of the overall training criterion (B.27) is equivalent to the following minimisation criterion:

$$\begin{aligned}\underset{q(\boldsymbol{z})}{\arg\min}\Big\{&\mathrm{KL}\big(q(\boldsymbol{v},\boldsymbol{z})||p(\boldsymbol{v},\boldsymbol{z})\big)+C\mathcal{R}\big(q(\boldsymbol{H},\boldsymbol{z})\big)+\\ &C_2\Big(\mathrm{KL}\big(q(\boldsymbol{v},\boldsymbol{z})||p(\boldsymbol{v},\boldsymbol{z})\big)-\sum_{\boldsymbol{z}}\int q(\boldsymbol{\Theta},\boldsymbol{z})\log p(\mathcal{D}|\boldsymbol{\Theta},\boldsymbol{z})d\boldsymbol{\Theta}\Big)\Big\}\end{aligned}\tag{B.28}$$

Distribution $q(\boldsymbol{z})$ can be factorised to $\prod_n q(z_n)$ as described in (B.18). Therefore, for the nth indicator z_n, the optimal distribution can obtained by minimising:

$$\begin{aligned}\underset{q(z_n)}{\arg\min}\Big\{&(1+C_2)\int\sum_{z_n}q(\boldsymbol{v})q(z_n)\log\frac{q(z_n)}{p(z_n|\boldsymbol{v})}d\boldsymbol{v}+C\mathcal{R}\big(q(\boldsymbol{H},z_n)\big)-\\ &C_2\sum_{z_n}\int q(\boldsymbol{\theta}_{z_n})q(z_n)\log p(\boldsymbol{x}_n|\boldsymbol{\theta}_{z_n})d\boldsymbol{\theta}_{z_n}\Big\}\end{aligned}\tag{B.29}$$

where $\mathcal{R}\big(q(\boldsymbol{H},z_n)\big)$ is the hinge loss function defined in (B.9) for the nth instance:

$$\begin{aligned}\mathcal{R}\big(q(\boldsymbol{H},&z_n)\big)=\Big[\max_{w\neq w_n}\Big\{\mathcal{L}(w,w_n)+\\ &\int\sum_{z_n}q(\boldsymbol{H})q(z_n)\Big(F(w,\boldsymbol{x}_n;\boldsymbol{H},z_n)-F(w_n,\boldsymbol{x}_n;\boldsymbol{H},z_n)\Big)d\boldsymbol{H}\Big\}\Big]_+\end{aligned}\tag{B.30}$$

Since the distribution $q(z_n)$ to be estimated is inside the maximisation, it is intractable to find the best competing hypothesis w. In order to make optimisation of $q(z_n)$ tractable, the hinge loss function (B.30) is relaxed to an upper bound by moving out the sum over z_n

from the maximisation[1]:

$$\mathcal{R}_{\text{up}}\big(q(\boldsymbol{H},z_n)\big) = \sum_{z_n} q(z_n)\Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}(w,w_n)+ \int q(\boldsymbol{H})\Big(F(w,\boldsymbol{x}_n;\boldsymbol{H},z_n)-F(w_n,\boldsymbol{x}_n;\boldsymbol{H},z_n)\Big)d\boldsymbol{H}\Big\}\Bigg]_+ \tag{B.31}$$

Since $\hat{q}(\boldsymbol{\eta}_m)$ (which was discussed in the previous section) is given, substituting the definition of the discriminant function (B.1) in, the hinge loss function (B.31) can be further written as:

$$\mathcal{R}_{\text{up}}\big(q(\boldsymbol{H},z_n)\big) = \sum_{z_n} q(z_n)\Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}(w,w_n)+\Big(\boldsymbol{\mu}_{z_n}^{\mathsf{T}}\phi(\boldsymbol{x}_n,w)-\boldsymbol{\mu}_{z_n}^{\mathsf{T}}\phi(\boldsymbol{x}_n,w_n)\Big)\Big\}\Bigg]_+ \tag{B.32}$$

where $\boldsymbol{\mu}_{z_n}$ is the mean of $\hat{q}(\boldsymbol{\eta}_{z_n})$. By using this relaxed criterion (B.32), the best competing hypothesis can be found. Thus, by relaxing the hinge loss function (B.30) to the upper bound, optimisation of $\hat{q}(z_n)$ becomes tractable.

Since the form of the distribution $q(z_n)$ is given, which is a categorical distribution with parameters $\boldsymbol{\varphi}_n = \{\varphi_{n,1},\ldots,\varphi_{n,M}\}$, estimation of $\hat{q}(z_n)$ is to estimate these parameters. Substituting $q(z_n)$, the hinge loss function $\mathcal{R}_{\text{up}}\big(q(\boldsymbol{H},z_n)\big)$ (given in (B.32)) and the prior distribution $p(z_n|\boldsymbol{v})$ (defined in (B.6) and (B.8)) in, the minimisation criterion (B.29) can be written as:

$$\begin{aligned}\arg\min_{\boldsymbol{\varphi}_n}\Bigg\{&(1+C_2)\sum_m\Big(\varphi_{n,m}\log\varphi_{n,m}-\varphi_{n,m}E_{q(v_m)}\big[\log v_m\big]- \\ &\varphi_{n,m}\sum_{i=1}^{m-1}E_{q(v_i)}\big[\log(1-v_i)\big]\Big)+C\sum_m\varphi_{n,m}\mathscr{F}_{n,m}- \\ &C_2\sum_m\varphi_{n,m}\Big(\log h(\boldsymbol{x})+E_{q(\boldsymbol{\theta}_m)}\big[\boldsymbol{\theta}_m\big]^{\mathsf{T}}\boldsymbol{x}-E_{q(\boldsymbol{\theta}_m)}\big[A(\boldsymbol{\theta}_m)\big]\Big)\Bigg\}\end{aligned} \tag{B.33}$$

where:

$$\mathscr{F}_{n,m} = \Bigg[\max_{w\neq w_n}\Big\{\mathcal{L}(w,w_n)+\Big(\boldsymbol{\mu}_m^{\mathsf{T}}\phi(\boldsymbol{x}_n,w)-\boldsymbol{\mu}_m^{\mathsf{T}}\phi(\boldsymbol{x}_n,w_n)\Big)\Big\}\Bigg]_+ \tag{B.34}$$

[1] An alternative approximate approach similar to [95] can be adopted. By using the optimal distribution $\hat{q}(z_n)$ in the last iteration, the best competing hypothesis can be obtained.

By minimising (B.33), the mth optimal parameter $\hat{\varphi}_{n,m}$ can be described as:

$$\hat{\varphi}_{n,m} \propto \exp\left\{\left(E_{q(v_m)}\left[\log v_m\right] + \sum_{i=1}^{m-1} E_{q(v_i)}\left[\log(1-v_i)\right]\right) - \frac{C}{1+C_2}\mathscr{F}_{n,m} + \frac{C_2}{1+C_2}\left(E_{q(\boldsymbol{\theta}_m)}\left[\boldsymbol{\theta}_m\right]^{\mathsf{T}}\boldsymbol{x} - E_{q(\boldsymbol{\theta}_m)}\left[A(\boldsymbol{\theta}_m)\right]\right)\right\} \tag{B.35}$$

Since $q(v_m) = \text{Beta}(c_{m,1}, c_{m,2})$ is a beta distribution (discussed at the beginning of section B.2), in (B.35) $E_{q(v_m)}\left[\log v_m\right] = \psi(c_{m,1}) - \psi(c_{m,1}+c_{m,2})$, and $E_{q(v_i)}\left[\log(1-v_i)\right] = \psi(c_{i,2}) - \psi(c_{i,1}+c_{i,2})$, where $\psi(\cdot)$ is the *digamma function*. $q(\boldsymbol{\theta}_m)$ is an exponential family distribution described in (B.15), and $A(\boldsymbol{\theta}_m)$ is a fixed function in the exponential family distribution. For the data that cannot be classified correctly, the hinge loss $\mathscr{F}_{n,m}$ defined in (B.34) tends to have greater value, and this leads to a smaller value of $\varphi_{n,m}$ given in (B.35). This means the term $\mathscr{F}_{n,m}$ biases the allocations of data towards the experts where they can be better classified.

B.3 Classification

In classification, given an input $\boldsymbol{x}$, the class label can be predicted by minimising the overall discriminant function of the iSVM described in equation (B.3):

$$\hat{w} = \arg\max_{w} \sum_{z} \int F(w, \boldsymbol{x}; \boldsymbol{H}, z)\hat{q}(\boldsymbol{H}, z)d\boldsymbol{H} \tag{B.36}$$

where z is the indicator variable corresponding to the input $\boldsymbol{x}$. The optimal distribution $\hat{q}(\boldsymbol{H}, z)$ is an approximation to $p(\boldsymbol{H}, z|\boldsymbol{x}, \mathcal{D})$, where $\mathcal{D}$ is the training data. This distribution $p(\boldsymbol{H}, z|\boldsymbol{x}, \mathcal{D})$ can be written as:

$$p(\boldsymbol{H}, z|\boldsymbol{x}, \mathcal{D}) = P(z|\boldsymbol{x}, \mathcal{D}, \boldsymbol{H})p(\boldsymbol{H}|\boldsymbol{x}, \mathcal{D}) \tag{B.37}$$

As discussed in section B.2, the optimal distribution $\hat{q}(\boldsymbol{H}, z)$ can be factorised: $\hat{q}(\boldsymbol{H}, z) = \hat{q}(\boldsymbol{H})\hat{q}(z)$. Assume the parameters of experts $\boldsymbol{H}$ only depends on the training data $\mathcal{D}$ (namely $p(\boldsymbol{H}|\boldsymbol{x}, \mathcal{D}) = p(\boldsymbol{H}|\mathcal{D})$), and the assignment of the input to which expert only depends on the gating network (namely $P(z|\boldsymbol{x}, \mathcal{D}, \boldsymbol{H}) = P(z|\boldsymbol{x}, \mathcal{D})$) [213]. Thus, the optimal distribution for the experts $\hat{q}(\boldsymbol{H})$ (an approximation to $p(\boldsymbol{H}|\mathcal{D})$) can be obtained in

training. The optimal distribution $\hat{q}(z)$ of the indicator needs to be inferred in classification, and variational inference can be applied:

$$\arg\min_{q(z)} \mathrm{KL}\big(q(z)||P(z|\boldsymbol{x},\mathcal{D})\big) \tag{B.38}$$

Since without the parameters for the gating network (namely $\boldsymbol{v}$ and $\boldsymbol{\Theta}$), the distribution over z is hard to evaluate. Thus, the minimisation criterion (B.38) is relaxed to an upper bound:

$$\arg\max_{q(z)} \mathrm{KL}\big(q(z)q(\boldsymbol{v})q(\boldsymbol{\Theta})||p(z,\boldsymbol{v},\boldsymbol{\Theta}|\boldsymbol{x},\mathcal{D})\big) \tag{B.39}$$

Then optimisation of $\hat{q}(z)$ becomes the same as standard variational inference for the infinite mixture model [19]. Given the optimal distributions $\hat{q}(\boldsymbol{v})$ and $\hat{q}(\boldsymbol{\Theta})$ which are obtained in the training phase, $\hat{q}(z)$ can be obtained by minimising (B.39). It is worth noting that the distributions $\hat{q}(\boldsymbol{v})$, $\hat{q}(\boldsymbol{\Theta})$ and $\hat{q}(z)$ do not need to be iteratively optimised as in inference of the infinite mixture model, since the optimal distributions $\hat{q}(\boldsymbol{v})$ and $\hat{q}(\boldsymbol{\Theta})$ are obtained in training and stay the same in classification. Thus, only one iteration needs to operated in optimising $\hat{q}(z)$ given $\hat{q}(\boldsymbol{v})$ and $\hat{q}(\boldsymbol{\Theta})$.

Appendix C

Hierarchical Dirichlet Processes

Many applications involve groups of data, and these groups are linked to each other. For example, documents are modelled as coming from an underlying set of topics, and these topics are shared across documents [21]; Haplotypes are shared among individuals in a population and across populations [53, 167] . A Dirichlet process can be adopted to model each group of the data. However, clusters cannot be shared cross groups, since the base distribution $\boldsymbol{G}_0$ is continuous. The solution is to use a common discrete base distribution. Rather than treating the base distribution parametrically, the base distribution can be treated non-parametrically as being sampled from a non-parametric model. In particularly, Dirichlet processes provide distributions over discrete distributions with wide support. When the base distribution $\boldsymbol{G}_0$ of a Dirichlet process itself is drawn from a Dirichlet process, this yields a hierarchical model called the *hierarchical Dirichlet process* (HDP) [174, 175, 176].

The HDP defines a groups of probability measures $\{\boldsymbol{G}_1, \cdots, \boldsymbol{G}_J\}$ and a global base measure $\boldsymbol{G}_0$. The global measure $\boldsymbol{G}_0$ is given a Dirichlet process with concentration parameter β and base distribution $\boldsymbol{Q}$:

$$\boldsymbol{G}_0 \sim \mathrm{DP}(\beta, \boldsymbol{Q}) \tag{C.1}$$

The group specific distributions $\boldsymbol{G}_j$ are independent to each other given $\boldsymbol{G}_0$, and they are sampled from a Dirichlet process with concentration parameter α and a common base mea-

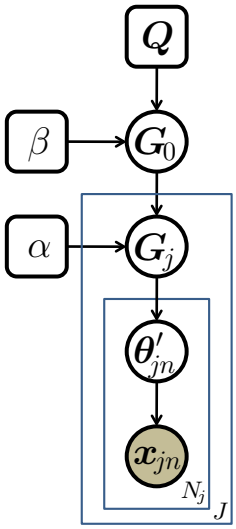


Figure C.1: *The graphical model of the hierarchial Dirichlet process mixture model.*

sure $\boldsymbol{G}_0$:

$$\boldsymbol{G}_j \sim \mathrm{DP}(\alpha, \boldsymbol{G}_0) \tag{C.2}$$

If $\boldsymbol{G}_j$ of different groups are expected to have different average variability from the base measure $\boldsymbol{G}_0$, the group specific concentration parameter α_j can be adopted to each group j [175]. This hierarchical model induces a sharing of atoms $\{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ among different random measures $\boldsymbol{G}_j$, since each inherits its set of atoms from the common base measure $\boldsymbol{G}_0$ [174]. The sharing of $\{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ can be illustrated by stick-breaking and Chinese restaurant franchise representations for the HDP, which will be discussed in the following sections.

Analogous to the infinite mixture model, groups of mixture models can be constructed based on the HDP. Given the sharing of atoms $\{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ induced by the HDP, the mixture models based on the HDP share parameters. Consider a set of data with J related groups $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^{J}$, where the jth group of data $\mathcal{D}_j = \{\boldsymbol{x}_{jn}\}_{n=1}^{N_j}$ has N_j observations. Each group is associated with a mixture model, and different groups are linked with each other by sharing the component parameters. The HDP provides a Bayesian non-parametric ap-

proach to model such grouped data:

$$\begin{aligned}
\boldsymbol{G}_0 &\sim \mathrm{DP}(\beta, \boldsymbol{Q}) \\
\boldsymbol{G}_j &\sim \mathrm{DP}(\alpha, \boldsymbol{G}_0) \\
\boldsymbol{\theta}'_{jn} &\sim \boldsymbol{G}_j \\
\boldsymbol{x}_{jn} &\sim p(\boldsymbol{x}|\boldsymbol{\theta}'_{jn})
\end{aligned} \tag{C.3}$$

This type of mixture model is called the *hierarchical Dirichlet process mixture model*. The corresponding graphical model of the HDP mixture model is illustrated in Figure C.1. In the HDP mixture model (C.3), given that $\boldsymbol{G}_j$ is discrete, samples $\{\boldsymbol{\theta}'_{j1}, \cdots, \boldsymbol{\theta}'_{jN_j}\}$ from $\boldsymbol{G}_j$ have positive probability taking identical values. This induces the cluster property of the mixture model for each group of data. These mixture models are based on Dirichlet processes with a common discrete base distribution $\boldsymbol{G}_0$. This leads to the sharing of parameters among these mixture models.

C.1 Stick-breaking Construction

In this section the stick-breaking construction of a hierarchical Dirichlet process (HDP) will be discussed. This construction gives an explicit representation of draws from a HDP, and provides insight into the sharing of atoms $\{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$ among different Dirichlet processes.

Given that the global base measure is distributed as a Dirichlet process $\boldsymbol{G}_0 \sim \mathrm{DP}(\beta, \boldsymbol{Q})$, it can be described in the form the stick-breaking representation of the Dirichlet process described in (4.27):

$$\begin{aligned}
\boldsymbol{c} &\sim \mathrm{GEM}(\beta) \\
\boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\
\boldsymbol{G}_0 &= \sum_{m=1}^{\infty} c_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)
\end{aligned} \tag{C.4}$$

where the weights $\boldsymbol{c} = \{c_m\}_{m=1}^{\infty}$ are obtained from the stick-breaking process described in (4.26). Given that each random measure $\boldsymbol{G}_j$ is also distributed as a Dirichlet process $\boldsymbol{G}_j \sim \mathrm{DP}(\alpha, \boldsymbol{G}_0)$, the corresponding stick-breaking construction for $\boldsymbol{G}_j$ can be described

as:

$$\begin{aligned}\boldsymbol{\pi}'_j &\sim \text{GEM}(\alpha)\\ \boldsymbol{\theta}''_{ji} &\sim \boldsymbol{G}_0\\ \boldsymbol{G}_j &= \sum_{i=1}^{\infty} \pi'_{ji}\delta(\boldsymbol{\theta}, \boldsymbol{\theta}''_{ji})\end{aligned} \tag{C.5}$$

where the weights $\boldsymbol{\pi}' = \{\pi'_i\}_{i=1}^{\infty}$ are obtained from the stick-breaking process (4.26). Since the base measure $\boldsymbol{G}_0$ is discrete, $\{\boldsymbol{\theta}''_{ji}\}_{i=1}^{\infty}$ from $\boldsymbol{G}_0$ have positive probability taking identical values.

Given that the random measure $\boldsymbol{G}_j$ is distributed according to a Dirichlet process $\boldsymbol{G}_j \sim \text{DP}(\alpha, \boldsymbol{G}_0)$, and the global base measure $\boldsymbol{G}_0$ has support at the atoms $\{\boldsymbol{\theta}_m\}_{m=1}^{\infty}$, $\boldsymbol{G}_j$ has support at these atoms as well. Thus the stick-breaking representation of $\boldsymbol{G}_j$ described in (C.5) can be rewritten as [174, 175]:

$$\boldsymbol{G}_j = \sum_{m=1}^{\infty} \pi_{jm}\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) \tag{C.6}$$

Assume the weights $\boldsymbol{\pi}_j = \{\pi_{jm}\}_{m=1}^{\infty}$ and $\boldsymbol{c} = \{c_m\}_{m=1}^{\infty}$ are probability measures on the discrete set $\{1, \cdots, \infty\}$. Given that $\boldsymbol{G}_j \sim \text{DP}(\alpha, \boldsymbol{G}_0)$ is distributed as a Dirichlet process on set $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_\infty\}$, the definition of the Dirichlet process (discussed in section 4.3.1) implies [175]:

$$\boldsymbol{\pi}_j \sim \text{DP}(\alpha, \boldsymbol{c}) \tag{C.7}$$

and the weights $\boldsymbol{\pi}_j$ can be obtained according the following stick-breaking construction:

$$\begin{aligned}v_{jm} &\sim \text{Beta}\left(\alpha c_m, \alpha\left(1 - \sum_{i=1}^{m} c_i\right)\right)\\ \pi_{jm} &= v_{jm}\prod_{i=1}^{m-1}(1 - v_{ji})\end{aligned} \tag{C.8}$$

where the weights $\boldsymbol{c} = \{c_m\}_{m=1}^{\infty}$ are from the stick-breaking representation for the global base measure $\boldsymbol{G}_0$ described in (C.4). Derivation of this construction (C.8) is discussed in detail in [175].

The HDP mixture model (C.3) was discussed in the previous section, in which each $\boldsymbol{\theta}'_{jn}$ is sampled from the random measure $\boldsymbol{G}_j$. According to the stick-breaking construction of

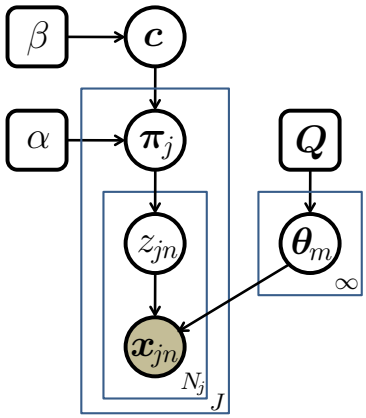


Figure C.2: *The graphical model of the stick-breaking construction of the hierarchial Dirichlet process mixture model.*

the HDP as discussed at the beginning of this section, $\boldsymbol{G}_j = \sum_{m=1}^{\infty} \pi_{jm} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)$, $\boldsymbol{\theta}_m \sim \boldsymbol{Q}$, and $\boldsymbol{\theta}'_{jn}$ takes the value $\boldsymbol{\theta}_m$ with probability π_{jm}. By introducing the indicator variable z_{jn}, the parameter $\boldsymbol{\theta}_m$ associated with the observation $\boldsymbol{x}_{jn}$ can be determined: $\boldsymbol{\theta}'_{jn} = \boldsymbol{\theta}_{z_{jn}}$. Thus an equivalent representation of the HDP mixture model can be described as [175]:

$$
\begin{aligned}
\boldsymbol{c} &\sim \text{GEM}(\beta) \\
\boldsymbol{\pi}_j &\sim \text{DP}(\alpha, \boldsymbol{c}) \\
z_{jn} &\sim \text{Categorical}(\boldsymbol{\pi}_j) \\
\boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\
\boldsymbol{x}_{jn} &\sim p(\boldsymbol{x} | \boldsymbol{\theta}_{z_{jn}})
\end{aligned}
\tag{C.9}
$$

This is the stick-breaking construction of the HDP mixture model, and the corresponding graphical model is illustrated in Figure C.2.

C.2 Chinese Restaurant Franchise

In section 4.3.3, the Chinese restaurant process (CRP) was discussed, which gives the predictive distribution of new observations by marginalising out the base distribution. Analogous to the CRP, by marginalising out the base distributions $\boldsymbol{G}_0$ and $\boldsymbol{G}_j$ in the hierarchical Dirichlet process (HDP), the *Chinese restaurant franchise* [175] can be resulted. In this in-

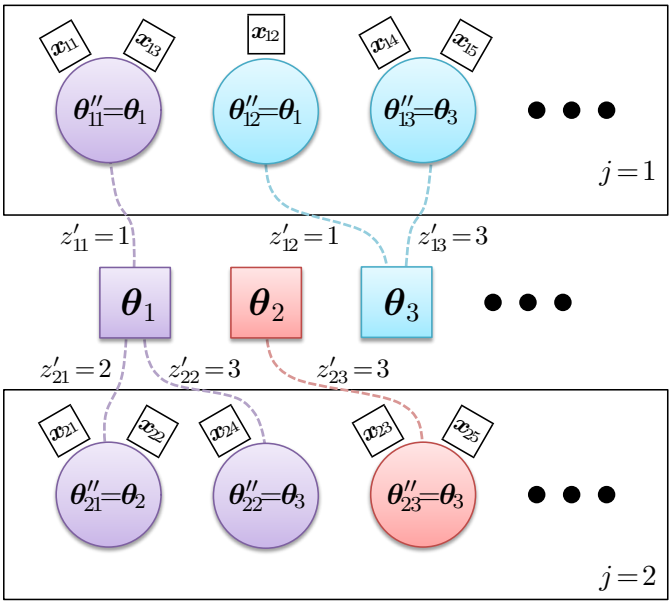


Figure C.3: *The Chinese restaurant franchise. This figure gives the current status of the Chinese restaurant franchise with two restaurants. When a new customer $\boldsymbol{x}_{16}$ come to the first restaurant, the probabilities of sitting at these three occupied tables are $\frac{2}{5+\alpha}$, $\frac{1}{5+\alpha}$ and $\frac{2}{5+\alpha}$ respectively, and the probability of choosing a new table is $\frac{\alpha}{5+\alpha}$. If this customer chooses a new table, the probabilities of choosing these three served dishes are $\frac{3}{6+\beta}$, $\frac{1}{6+\beta}$ and $\frac{2}{6+\beta}$ respectively, and the probability of choosing a dish $\boldsymbol{\theta}_m$ from the menu is $\frac{\beta}{6+\beta}$.*

terpretation, the CRP metaphor is extended to multiple restaurants which share a set of dishes.

In the Chinese restaurant franchise metaphor, a group of restaurants are defined, and in each restaurant customers (observations) $\{\boldsymbol{x}_{jn}\}_{n=1}^{N_j}$ sit at tables (components) $\{t_{jn}\}_{n=1}^{N_j}$. Each table serves a single dish (parameter) $\boldsymbol{\theta}''_{jt}$ shared by customers, and the dish is ordered from a global menu $\boldsymbol{G}_0$ shared cross restaurants. Let z'_{jt} denote the global parameter $\boldsymbol{\theta}_{z'_{jt}}$ is assigned to table t in restaurant j, and $\boldsymbol{z}'_j = \{z'_{j1}, z'_{j2}, \cdots\}$ give the assignments of dishes for all the tables in restaurant j. Analogous to the derivation of the Chinese restaurant process discussed in section 4.3.3, by integrating over $\boldsymbol{G}_0$ and $\boldsymbol{G}_j$, the conditional distribution

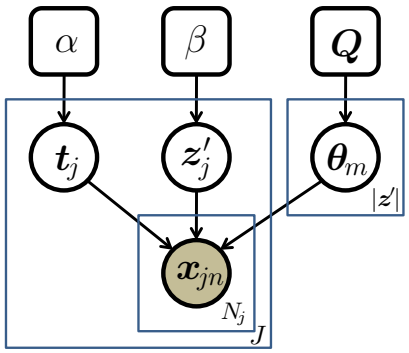


Figure C.4: *The graphical model of the hierarchial Dirichlet process mixture model based on the Chinese restaurant franchise.* $\boldsymbol{t}_j = \{t_{jn}\}_{n=1}^{N_j}$ *are table indicators for customers in restaurant* j*, and* N_j *is the number customers in that restaurant.* $\boldsymbol{z}'_j = \{z'_{jt}\}_{t=1}^{T_j}$ *are dish indicators for tables in restaurant* j*, and* T_j *is the number of occupied tables in that restaurant. These sets of indicators* $\boldsymbol{t}_j$ *and* $\boldsymbol{z}'_j$ *are generated by the Chinese restaurant franchise.* $|\boldsymbol{z}'|$ *is the number of unique values in set* $\boldsymbol{z}' = \{\boldsymbol{z}'_j\}_{j=1}^{J}$*.*

of the indicator variables t_{jn} and z'_{jt} can be described as [168, 175]:

$$P(t_{jn}|t_{j1}, \cdots, t_{jn-1}, \alpha) \propto \sum_{t=1}^{T_j} N_{jt}\delta(t_{jn}, t) + \alpha\delta(t_{jn}, T_j + 1) \tag{C.10}$$

$$P(z'_{jt}|\boldsymbol{z}'_1, \cdots, \boldsymbol{z}'_{j-1}, z'_{j1}, \cdots z'_{jt-1}, \beta) \propto \sum_{m=1}^{M} M_k\delta(z'_{jt}, m) + \beta\delta(z'_{jt}, M + 1) \tag{C.11}$$

where T_j is the number of currently occupied tables in restaurant j, N_{jt} is the number of customers sitting at table t in restaurant j, M is the total number of unique dishes served in all restaurants, and M_k is the number of tables served with dish $\boldsymbol{\theta}_k$ in all restaurants. Similar to the CRP, when a customer come to a restaurant, the probability of sitting at an occupied table is proportional to the number of people already sitting there, and the probability of sitting at a new table is proportional to α as described in (C.10). When a new table is chosen, the probability of choosing a served dish is proportional to the number of tables served with that dish, and the probability of choosing a new dish is proportional to β as described in (C.11). This Chinese restaurant franchise metaphor is illustrated in Figure C.3.

As discussed in the previous section, a HDP mixture model can be built based on the stick-breaking construction. Analogously, a HDP mixture model also can be based on the Chinese restaurant franchise. The graphical model of this HDP mixture model is illustrated in Figure C.4. Given the groups of indicators $\{t_{jn}\}_{n=1}^{N_j}$ and $\{z'_{jt}\}_{t=1}^{T_j}$ with $j \in \{1, 2, \cdots\}$

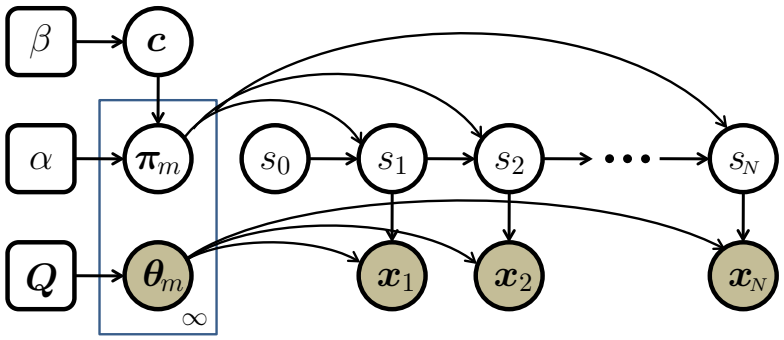


Figure C.5: *The graphical model of the infinite hidden Markov model.*

(where N_j is the number observations in group j and T_j is the number of unique values in set $\{t_{jn}\}_{n=1}^{N_j}$) from the Chinese restaurant franchise described in (C.10) and (C.11), the corresponding generative process of the HDP mixture model can be described as follows:

$$\begin{aligned} \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ \boldsymbol{x}_{jn} &\sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z'_{jt_{jn}}}) \end{aligned} \tag{C.12}$$

C.3 Relationships with Infinite HMMs

In the previous sections, the hierarchical Dirichlet process (HDP) and the mixture models based on this process were discussed. This section will discuss the *infinite hidden Markov model* [1] which is a specification of the HDP.

In a hidden Markov model (HMM), a sequence of the state variables $\{s_1, \cdots, s_N\}$ are linked through the state transition matrix, and the observations $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ are drawn independently conditional on the given states. In the HMM, the probability of generating next observation $\boldsymbol{x}_{n+1}$ conditional on the current state s_n is a finite mixture model, namely $p(\boldsymbol{x}_{n+1}|s_n) = \sum_{s_{n+1}} P(s_{n+1}|s_n)p(\boldsymbol{x}_{n+1}|s_{n+1})$, in which the component distribution is the distribution of the emitting state $p(\boldsymbol{x}_{n+1}|s_{n+1})$. Thus, the HMM is a dynamic variant of the finite mixture model. It would be interesting to generalise the finite mixture underlying the HMM to a Dirichlet process, and the resulting HMM with infinite number of states is called the *infinite HMM*, or the *HDP-HMM* [175].

[1] A hidden Markov model (HMM) with infinite number of states is called the infinite HMM in this thesis.

In the HMM, the current state s_n indexes one row of the transition matrix, and the probability of that row is the mixing proportion for choosing the next state s_{n+1}. The observation $\boldsymbol{x}_n$ is drawn from the component indexed by s_n (each state of the HMM corresponds a component). There are infinite number of states in the non-parametric counterpart of the HMM, and the Dirichlet process is employed to give the transition probability for each state. Moreover, these Dirichlet processes must be linked with each other, since any state needs to be reachable from other states. Thus the framework of the HDP is ideal for these linked Dirichlet processes. This results a HMM with infinite number of states, which is called the infinite HMM. The graphical model of the infinite HMM is illustrated in Figure C.5, and the corresponding generative process can be described as follows:

$$\begin{aligned} \boldsymbol{c} &\sim \mathrm{GEM}(\beta) \\ \boldsymbol{\pi}_m &\sim \mathrm{DP}(\alpha, \boldsymbol{c}) \\ s_n | s_{n-1} &\sim \mathrm{Categorical}(\boldsymbol{\pi}_{s_{n-1}}) \\ \boldsymbol{\theta}_m &\sim \boldsymbol{Q} \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x} | \boldsymbol{\theta}_{s_n}) \end{aligned} \tag{C.13}$$

where $\boldsymbol{\pi}_m^{\mathsf{T}}$ is the mth row of the transition matrix for the infinite HMM.

As discussed in this section, each state of the infinite HMM has a single component. More generally, the state distribution can be extended to be a mixture model (with infinite number of components), by introducing a latent variable z_n, that denotes which component the observation is associated with, for each state. In practice, a special treatment of the transition probability also can be applied to avoid rapid switching among the redundant states in the infinite HMM. These specifications of infinite HMMs are discussed in section 4.4.

Appendix D

Beta Processes

In the previous chapters, Dirichlet processes, hierarchical Dirichlet processes and the mixture models based these processes were discussed. For these mixture models, it is assumed that the observations can be partitioned into a discrete set of clusters, and each observation is assigned to a single cluster. This is particular clear in the Chinese restaurant process (CRP), in which each customer is associated with a single dish. It is more appropriate to assume each observation is associated with a collection of attributes, e.g. an animal can be both terrestrial and oviparous. This assumption makes each observation in a model can be assigned to a subset of clusters. The *Indian buffet process* [72] satisfies this assumption, in which each customer is associated with a subset of infinite dishes. Analogous to the Dirichlet process underlying the CRP, the underlying process of the India buffet process is a Beta process, which will be discussed in this chapter.

The beta process was first introduced by Hjort [86] for survival analysis. It is a *Lévy process* [12, 155] which is a right continuous stochastic process with stationary and independent increments. Formally, the beta process can be defined as follows. Let $\boldsymbol{\Theta}$ be a set, $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$ its σ-algebra, $\boldsymbol{B}_0$ a continuous probability measure on the measurable space $(\boldsymbol{\Theta}, \boldsymbol{\Sigma}_{\boldsymbol{\Theta}})$ and c a positive scalar. Then for any disjoint infinitesimal partition $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_L\}$ of $\boldsymbol{\Theta}$, the beta process is generated as follows [135]:

$$\boldsymbol{B}(\boldsymbol{A}_l) \sim \text{Beta}\Big(c\boldsymbol{B}_0(\boldsymbol{A}_l), c\big(1 - \boldsymbol{B}_0(\boldsymbol{A}_l)\big)\Big) \tag{D.1}$$

with $L \to \infty$ and $\boldsymbol{B}_0(\boldsymbol{A}_l) \to 0, \forall l \in \{1, \cdots, L\}$. This is a *beta process*, and denoted as:

$$\boldsymbol{B} \sim \text{BP}(c, \boldsymbol{B}_0) \tag{D.2}$$

In (D.1) Beta$(\cdot)$ is a beta distribution. Because of the aggregation property [1] of Dirichlet distributions, a Dirichlet process can be defined in terms of finite-dimensional distributions appealing to the *Kolmogorov consistency theorem*, which guarantees that a suitably consistent collection of finite-dimensional distributions will define a stochastic process. However, the sum of two beta variables is not beta distributed (then the Kolmogorov consistency theorem condition is not satisfied), hence the beta process is defined in the infinitesimal limit rather than being defined based on finite-dimensional probabilities. An alternative definition of the beta process based on the framework of completely random measures is discussed in [97, 178].

The random measure $\boldsymbol{B}$ from a beta process is a *completely random measure*[2] [102], and it can be described as:

$$\boldsymbol{B} = \sum_{m=1}^{\infty} p_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) \tag{D.3}$$

where $\delta(\cdot)$ is a Dirac delta function, each weight p_m satisfies $0 \leq p_m \leq 1$, and the sum of all weights is finite (the consequence of Campbell's theorem) [97] . The random measure $\boldsymbol{B}$ from a beta process is not a probability mass function as the measure from a Dirichlet process, but it can serve as the parameters of a Bernoulli process. The Bernoulli process based on this random measure $\boldsymbol{B}$ can be described as follows. Let $\boldsymbol{z}_n = \{z_{nm}\}_{m=1}^{\infty}$ be an infinite set of binary indicators (with values 0 or 1), and each z_{nm} corresponding to atom $\boldsymbol{\theta}_m$ be given by a bernoulli distribution:

$$z_{nm} \sim \text{Bernoulli}(p_m) \tag{D.4}$$

where $\{p_m\}_{m=1}^{\infty}$ are weights from the random measure $\boldsymbol{B}$ described in (D.3). Then the newly generated measure $\boldsymbol{Z}_n = \sum_{m=1}^{\infty} z_{nm} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)$ is drawn from a *Bernoulli process*, which is denoted as:

$$\boldsymbol{Z}_n \sim \text{BeP}(\boldsymbol{B}) \tag{D.5}$$

[1] The aggregation property of Dirichlet distributions can be described as: If $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) \sim \text{Dirichlet}(\alpha_1, \cdots, \alpha_L)$, then $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_i + \boldsymbol{x}_j, \cdots, \boldsymbol{x}_L) \sim \text{Dirichlet}(\alpha_1, \cdots, \alpha_i + \alpha_j, \cdots, \alpha_L)$.

[2] For a random measure $\boldsymbol{B}$ on the measurable space $(\boldsymbol{\Theta}, \boldsymbol{\Sigma}_{\boldsymbol{\Theta}})$, if the random variables $\boldsymbol{B}(\boldsymbol{A}_i)$ and $\boldsymbol{B}(\boldsymbol{A}_j)$ are independent for any disjoint sets $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ in $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$, $\boldsymbol{B}$ is a completely random measure.

Conjugacy Let $\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n$ be a collection of random measures from the Bernoulli process $\text{BeP}(\boldsymbol{B})$. Given $\boldsymbol{B}$, random measures $\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n$ are conditional independent. The posterior distribution of $\boldsymbol{B}$ given $\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n$ is still a beta process [174]:

$$\boldsymbol{B}|\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n, c, \boldsymbol{B}_0 \sim \text{BP}\Big(c+n, \frac{c}{c+n}\boldsymbol{B}_0 + \frac{1}{c+n}\sum_{i=1}^{n}\boldsymbol{Z}_i\Big) \tag{D.6}$$

This results from the conjugacy between the beta process and the Bernoulli process.

D.1 Indian Buffet Processes

In section 4.3.3, the Chinese restaurant process (CRP) was discussed. The CRP can be derived by integrating out the Dirichlet process in the predictive distribution (4.29). Analogously, the *Indian buffet process* (IBP) [72] can be motivated by integrating out the beta process (D.6) in the following predictive distribution:

$$\begin{aligned} P(\boldsymbol{Z}_{n+1}|\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n, c, \boldsymbol{B}_0) &= \int P(\boldsymbol{Z}_{n+1}|\boldsymbol{B})p(\boldsymbol{B}|\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n, c, \boldsymbol{B}_0)d\boldsymbol{B} \\ &= \text{BeP}\Big(\frac{c}{c+n}\boldsymbol{B}_0 + \sum_{m=1}^{\infty}\frac{N_{nm}}{c+n}\delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m)\Big) \end{aligned} \tag{D.7}$$

where N_{nm} is the number of indicators corresponding to $\boldsymbol{\theta}_m$ equaling 1, namely $N_{nm} = \sum_{i=1}^{n} z_{im}$ (note z_{im} is binary). Each z_{im} is from a Bernoulli distribution described in (D.4) which defines a Bernoulli process.

The IBP can be described as follows. Consider a restaurant with a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish until Poisson(γ)[1] number of dishes have been taken, where $\gamma = \boldsymbol{B}_0(\boldsymbol{\Theta})$ is the total mass of $\boldsymbol{B}_0$ with finite value. The $(n+1)$th customer moves along the buffet, for each previously sampled dish, the customer samples the dish with probability $\frac{N_{nm}}{c+n}$, where N_{nm} is the number of people having already token it. After the end of all previously sampled dishes is reached, the $(n+1)$th customer then tries Poisson($\frac{c\gamma}{c+n}$) number of new dishes [78]. A binary matrix sampled from a IBP is illustrated in Figure D.1.

To connect the IBP with (D.7), assume the total mass of $\boldsymbol{B}_0$ is finite, namely $\boldsymbol{B}_0(\boldsymbol{\Theta}) = \gamma$. The first customer corresponds to the measure $\boldsymbol{Z}_1$ which is distributed as a Bernoulli

[1] Poisson(γ) is a Poisson distribution with parameter γ.

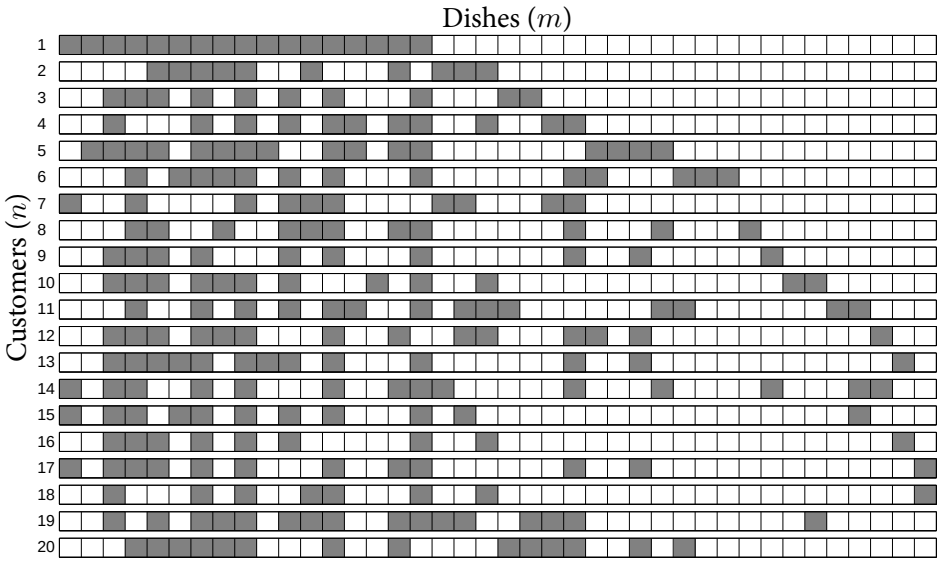


Figure D.1: *A binary matrix generated by the Indian buffet process with $\gamma = 10$. This figure is taken from [78].*

process $\boldsymbol{Z}_1 \sim \text{BeP}(\boldsymbol{B}_0)$. Then the sum of this Bernoulli process outcomes is distributed as a Poisson process with parameter $\boldsymbol{B}_0$. Thus, $\boldsymbol{Z}_1(\boldsymbol{\Theta}) \sim \text{Poisson}\big(\boldsymbol{B}_0(\boldsymbol{\Theta})\big) = \text{Poisson}(\gamma)$. This means the first customer tries Poisson(γ) number of dishes. The $(n+1)$th customer corresponds $\boldsymbol{Z}_{n+1}$, and it is distributed according to the Bernoulli process described in (D.7), which is the sum of two independent Bernoulli process. Analogous to $\boldsymbol{Z}_1$, the $(n+1)$th customer takes dish $\boldsymbol{\theta}_m$ with probability $\frac{N_{nm}}{c+n}$, and takes Poisson$(\frac{c\gamma}{c+n})$ number of new dishes [178].

D.2 Stick-breaking Constructions

In section 4.3.2, the stick-breaking construction of the Dirichlet process was discussed. This construction characterises the draws from Dirichlet processes as discrete random measures. As described in (D.6), a draw from the beta process is discrete with probability one, since the continue part $\frac{c}{c+n}\boldsymbol{B}_0$ becomes 0 when $n \to \infty$. This makes the existence of a stick-breaking construction for the beta process possible. As discussed in [178], such construction exists and a truncated representation can be described as follows:

$$\boldsymbol{B}_N = \sum_{n=1}^{N} \sum_{m=1}^{M_n} p_{nm} \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_{nm}) \tag{D.8}$$

where

$$\begin{aligned} M_n &\sim \text{Poisson}\Big(\frac{c}{c+n-1}\gamma\Big) \\ p_{nm} &\sim \text{Beta}(1, c+n-1) \\ \boldsymbol{\theta}_{nm} &\sim \frac{1}{\gamma}\boldsymbol{B}_0 \end{aligned} \tag{D.9}$$

In (D.9), $\gamma = \boldsymbol{B}_0(\boldsymbol{\Theta})$ is the total mass on $\boldsymbol{\Theta}$. Equation (D.8) gives a truncated representation of the stick-breaking construction for the beta process. When $N \to \infty$, this truncated representation $\boldsymbol{B}_N$ converges to $\boldsymbol{B}$ with probability one [178].

A stick-breaking construction of the beta process based on the Lévy measure is discussed in [174]. When $c = 1$, this stick-breaking construction can be simplified as:

$$\boldsymbol{B}_M = \sum_{m=1}^{M} p_m \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_m) \tag{D.10}$$

where

$$\begin{aligned} v_m &\sim \text{Beta}(1, \gamma) \\ p_m &\sim \prod_{i=1}^{m}(1 - v_m) \\ \boldsymbol{\theta}_m &\sim \frac{1}{\gamma}\boldsymbol{B}_0 \end{aligned} \tag{D.11}$$

Again, $\gamma = \boldsymbol{B}_0(\boldsymbol{\Theta})$ is the total mass on $\boldsymbol{\Theta}$. When $M \to \infty$, this truncated representation $\boldsymbol{B}_M$ converges to $\boldsymbol{B}$. Compared with the stick-breaking construction of the Dirichlet process described in (4.26), both constructions employ the same breaking ratio $v_m \sim \text{Beta}(1, \gamma)$. For the Dirichlet process, the parts broken off are used, whereas the remaining parts are used for the beta process. Thus, in Figure 4.2 the blue parts (remaining parts) of the stick correspond the weights $\{p_1, p_2, \ldots\}$ from a beta process.

Analogous to the mixture models based on Dirichlet processes, mixture models can be based on beta processes, e.g. the infinite overlapping mixture model based on the Indian buffet process [83].

Appendix E

Large Margin Training for the Experts

In section 5.3.3, large margin training is applied to each expert in training the infinite log-linear model given a sampled set of indicator variables. In this appendix, all the experts of the infinite log-linear model will be treated as a single set of parameters, and an overall large margin training criterion for all experts (log-linear models) will be discussed. This large margin training criterion is a specific example of the general criterion discussed in section 5.2. This appendix also shows that large margin training for each expert can be viewed as a special example of large margin training for all the experts.

E.1 The Training Criterion

In the infinite structured discriminative model described in (5.20), the parameters of all the experts are $\boldsymbol{H} = \{\boldsymbol{\eta}_m\}_{m=1}^{\infty}$. Assume there are N training instances $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, the posterior distribution of $\boldsymbol{H}$ can be estimated according to the large margin training criterion described in (5.11), which is the general criterion discussed in section 5.2 with a hinge loss criterion function:

$$\begin{aligned} &\arg\min_{q(\boldsymbol{H})} \left\{ \mathrm{KL}\big(q(\boldsymbol{H})||p(\boldsymbol{H})\big) - \int q(\boldsymbol{H}) \mathcal{F}(\boldsymbol{H}; \mathcal{D}) d\boldsymbol{H} \right\} \\ &\text{s.t.} \quad q(\boldsymbol{H}) \in \mathcal{P}_{\text{prob}} \end{aligned} \tag{E.1}$$

where $p(\boldsymbol{H})$ is the prior distribution of $\boldsymbol{H}$, and $\mathcal{F}(\boldsymbol{H};\mathcal{D})$ is a hinge loss function described in (5.9):

$$\mathcal{F}(\boldsymbol{H};\mathcal{D}) = -\sum_{n=1}^{N}\left[\max_{W,\rho\neq W_n,\rho_n}\left\{\mathcal{L}(W,W_n)-\mathcal{M}(W,W_n;\boldsymbol{H},\boldsymbol{O}_n)\right\}\right]_+ \tag{E.2}$$

In (E.2), for each training instance, the most competing hypothesis and segmentation pair (W,ρ) is found over all possible hypotheses and segmentations except the reference with the corresponding segmentation (W_n,ρ_n), and ρ_n is the most likely segmentation obtained by the HMM as described in equation (5.25). $\mathcal{L}(W,W_n)$ is the loss between the hypothesis W and the reference W_n, and $[\cdot]_+$ is the hinge-loss defined in (5.10). $\mathcal{M}(W,W_n;\boldsymbol{H},\boldsymbol{O}_n)$ is the margin, which determines how well the reference label W_n can be correctly separated with the hypothesis W [165]. In this work the definition of the margin is:

$$\mathcal{M}(W_n,W;\boldsymbol{H},\boldsymbol{O}_n) = \int\sum_{\boldsymbol{z}}\log\frac{P(W_n,|\boldsymbol{O}_n,\boldsymbol{H},z_n)}{P(W|\boldsymbol{O}_n,\boldsymbol{H},z_n)}q(\boldsymbol{\Theta},\boldsymbol{z})d\boldsymbol{\Theta} \tag{E.3}$$

where the conditional probability $P(W|\boldsymbol{O},\boldsymbol{H},z)$ is given by the log-linear model defined in (5.24). Note, the denominator terms of the log-linear model can be cancelled out in this margin definition (E.3). With this margin definition, the resulting margin has a form similar to the margin in the structured SVM. Thus, the efficient training approaches used in the structured SVM can be implemented, e.g. applying the Viterbi algorithm to find the best competing hypothesis. Other definitions of the margin will be discussed in the following section.

E.1.1 Other Margin Definitions

In a large margin training criterion, the margin $\mathcal{M}(\cdot)$ determines how well the reference label W_n can be correctly separated with the hypothesis W [165]. Thus, the definition of the margin should be closely related to the class posterior distribution in classification. One option for the definition of the margin is the logarithm of the class posterior distribution ratio:

$$\mathcal{M}(W_n,W;\boldsymbol{O}_n) = \log\frac{P(W_n|\boldsymbol{O}_n,\mathcal{D})}{P(W|\boldsymbol{O}_n,\mathcal{D})} \tag{E.4}$$

where the distribution $P(W|\boldsymbol{O},\mathcal{D})$ is the class posterior distribution used in classification, e.g. the class posterior distribution $P(W|\boldsymbol{O},\mathcal{D}) = \int P(W|\boldsymbol{O},\mathcal{G})p(\mathcal{G}|\mathcal{D})d\mathcal{G}$ in (5.26).

This is a direct way to define the margin, in which the class posterior distribution gives the probabilities of the observation having different labels. This margin gives a score which can measure how well the data can be correctly separated. In terms of the margin defined in (E.4), the class posterior distribution $P(W|\boldsymbol{O}, \mathcal{D})$ is given by an infinite mixture of log-linear models, and the denominator term of each log-linear model is different. Thus, the denominator terms cannot be cancelled out. This leads to inefficiency in training.

Alternatively, the margin can be defined as the expectation of the log posterior ratio:

$$\mathcal{M}(W_n, W; \boldsymbol{O}_n) = \int \sum_{\boldsymbol{z}} \log \frac{P(W_n, |\boldsymbol{O}_n, \boldsymbol{H}, z_n)}{P(W|\boldsymbol{O}_n, \boldsymbol{H}, z_n)} q(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) d(\boldsymbol{\Theta}, \boldsymbol{H}) \tag{E.5}$$

where the conditional probability $P(W|\boldsymbol{O}, \boldsymbol{H}, z)$ is given by a structured discriminative model, e.g. the log-linear model defined in (5.24). This type of margin definition has the same form as the margin defined in maximum entropy discrimination (MED) [93]. This margin (E.5) is also related to the margin (E.3) for the general criterion (E.1), where the integral over the parameters of the experts $\boldsymbol{H}$ are outside the hinge loss function.

In the structured SVM, which can be interpreted as a log-linear model with large margin training [210], the margin is defined the logarithm of the conditional probabilities ratio. Analogously, the margin based on a single expert z_n can be defined as:

$$\mathcal{M}(W_n, W; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = \log \frac{P(W_n|\boldsymbol{O}_n, \boldsymbol{H}, z_n)}{P(W|\boldsymbol{O}_n, \boldsymbol{H}, z_n)} \tag{E.6}$$

When the conditional probability $P(W|\boldsymbol{O}, \boldsymbol{H}, z)$ is given by the log-linear model defined in (5.24), the denominator terms of the log-linear models can be cancelled out in this margin definition (E.6). This margin then can be described in the form a linear function of $\boldsymbol{\eta}_{z_n}$:

$$\mathcal{M}(W_n, W; \boldsymbol{O}_n, \boldsymbol{H}, z_n) = \boldsymbol{\eta}_{z_n}^{\mathsf{T}} \Big(\Phi(\boldsymbol{O}_n, W_n, \rho_n) - \Phi(\boldsymbol{O}_n, W, \rho) \Big) \tag{E.7}$$

E.2 Large Margin Training

As discussed in section 5.3, from a Bayesian perspective, the model parameters are random variables, which are marginalised out in classification. In the infinite log-linear model, $\boldsymbol{H}$ is the parameter set of all the experts, and the size of the set $\boldsymbol{H}$ is infinite. Moreover, the maximisation is inside the integral in the large margin training criterion (E.1). It is computationally intractable to solve this general criterion directly. In order to make it tractable, an

Algorithm 6: The training procedure of the infinite log-linear model

Initialise: $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ and $\hat{q}(\boldsymbol{H})$

repeat

1. Given $\hat{q}(\boldsymbol{H})$, update the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$[1]described in (E.10).
2. Given the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, update $\hat{q}(\boldsymbol{H})$ by minimising the large margin training criterion (E.1) with the hinge loss criterion function (E.2).

until *converge or maximum number of iteration is reached*;

approximation is made here. As discussed in section 5.2, the large margin training criterion becomes tractable when the distribution $q(\boldsymbol{H})$ to be estimated is a Dirac delta function with parameters $\hat{\boldsymbol{H}} = \{\hat{\boldsymbol{\eta}}_m\}_{m=1}^{\infty}$:

$$q(\boldsymbol{H}) = \delta(\boldsymbol{H}, \hat{\boldsymbol{H}}) = \prod_{m=1}^{\infty} \delta(\boldsymbol{\eta}_m, \hat{\boldsymbol{\eta}}_m) \tag{E.8}$$

The optimal distribution for the whole model is assumed to be:

$$\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \approx \hat{q}(\boldsymbol{H})\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z}) \tag{E.9}$$

where the overall distribution is decomposed into two distributions. One distribution $\hat{q}(\boldsymbol{H})$ is for the experts, the second $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ is for the gating network. These two distributions are dependent on each other, and in this work the dependency is restricted to a specific form: The optimal distribution $\hat{q}(\boldsymbol{H})$ for the expert is estimated by minimising the large margin training criterion (E.1) given $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$; And the optimal distribution for the gating network is defined as the following posterior distribution given $\hat{q}(\boldsymbol{H})$:

$$\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z}) = \int p(\boldsymbol{\Theta}, \boldsymbol{z}|\boldsymbol{H}, \mathcal{D})\hat{q}(\boldsymbol{H})d\boldsymbol{H} = p(\boldsymbol{\Theta}, \boldsymbol{z}|\hat{\boldsymbol{H}}, \mathcal{D}) \tag{E.10}$$

This is the posterior distribution for the gating network, and can be obtained through Bayesian rule. Since the forms of the distributions $\hat{q}(\boldsymbol{H})$ and $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ are dependent on each other, the optimal distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z})$ is obtained by alternatively optimising distributions $\hat{q}(\boldsymbol{H})$ and $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ as described in Algorithm 6.

For the infinite log-linear model, the posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{z}|\hat{\boldsymbol{H}}, \mathcal{D})$ in (E.10) does not have a closed form, so Monte Carlo approaches must again be applied. Here Gibbs sampling is used to iteratively draw from the conditional posterior distribution of each parameter in turn. These conditional posterior distributions (conditional posterior distributions for each component and indicator $p(\boldsymbol{\theta}_m|\boldsymbol{z}^{(k)}, \hat{\boldsymbol{H}}, \mathcal{D})$ and $P(z_n|\boldsymbol{z}_{-n}^{(k)}, \boldsymbol{\Theta}^{(k-1)}, \hat{\boldsymbol{H}}, \mathcal{D})$)

are the same as the distributions detailed in sections 5.3.1.3 and 5.3.1.4. The optimal distribution of the experts' parameters $\hat{q}(\boldsymbol{H})$ is estimated according to the large margin training criterion (E.1) with hinge loss criterion function (E.2). Thus estimation of $\hat{q}(\boldsymbol{H})$ depends on $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, and this dependance is shown in the margin definition (E.3). The distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ depends on $\hat{q}(\boldsymbol{H})$ in turn as shown in equation (E.10), hence the training procedure can be summarised as an iterative process described in Algorithm 6. Theoretically, there is no guarantee on the convergency of this iterative training. If it converges, the optimal distribution is the distribution that minimises the large margin training criterion (E.1), and also gives the standard conditional posterior distribution (in Bayesian inference) for the gating network as described in (E.10). In the following sections, estimation of $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ and $\hat{q}(\boldsymbol{H})$ will be discussed.

E.2.1 Estimation of $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$

The optimal distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ is the posterior distribution of the parameters of the gating network $p(\boldsymbol{\Theta}, \boldsymbol{z}|\hat{\boldsymbol{H}}, \mathcal{D})$ as described in equation (E.10). This distribution is the same as the conditional posterior distribution of the parameters for the gating network in Bayesian inference of the infinite log-linear model discussed in section 5.3.1.2. Since this posterior distribution does not have a closed form, Gibbs sampling is applied to sample from the conditional posterior distribution of each parameter, and the process of sampling from these conditional posterior distributions is the same as that in sections 5.3.1.3 and 5.3.1.4.

E.2.2 Estimation of $\hat{q}(\boldsymbol{H})$

As discussed in the previous section, the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ does not have a closed form. Here Gibbs sampling is used to sample from this distribution. The margin (E.3) is defined as integrating this distribution, hence the margin can be approximated by summing over K samples:

$$\mathcal{M}_2(W_n, W; \boldsymbol{H}, \boldsymbol{O}_n) \approx \frac{1}{K} \sum_{k=1}^{K} \log \frac{P(W_n, |\boldsymbol{O}_n, \boldsymbol{H}, z_n^{(k)})}{P(W|\boldsymbol{O}_n, \boldsymbol{H}, z_n^{(k)})} \tag{E.11}$$

[1] In training, the distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$ is approximated by samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}\}_{k=1}^{K}$.

where the samples $\boldsymbol{z}^{(k)} = \{z_1^{(k)}, \ldots, z_N^{(k)}\}$ are drawn from $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, and can be obtained from the first step in Algorithm 6, which was discussed in section E.2.1. By substituting the definition of the log-linear model (5.24) into the margin described in (E.11), the denominator terms of the log-linear models can be cancelled out. Then, the margin (E.11) can be expressed as:

$$\mathcal{M}_2(W_n, W; \boldsymbol{H}, \boldsymbol{O}_n) \approx \frac{1}{K} \sum_{k=1}^{K} \left(\boldsymbol{\eta}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n, \rho_n) - \boldsymbol{\eta}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W, \rho) \right) \tag{E.12}$$

where ρ is the most likely segmentation (corresponding to sentence W) obtained by the HMM as described in equation (5.25).

In the infinite SVM proposed by Zhu [213], each expert is a discriminant function[1], which is defined as a linear function of $\boldsymbol{\eta}$, namely $F = \boldsymbol{\eta}^{\mathsf{T}} \Phi(\boldsymbol{O}, W, \rho)$. As the margin is defined as the difference of the linear functions, the margin is also linear. In the training criterion of the infinite SVM described in Appendix B.1, the integral (over $\boldsymbol{\eta}$) is inside the maximisation. Thus the integrand is a linear function of $\boldsymbol{\eta}$, and the integral over $\boldsymbol{\eta}$ is tractable in the infinite SVM.

For the infinite structured discriminative model discussed in this appendix, the experts are log-linear models, and the margin is defined as the expectation of the log-posterior ratio (E.3). With this margin definition, the denominator terms of the log-linear models are cancelled. As discussed at the beginning of section E.2, the distribution for the experts' parameters is assumed to be a Dirac delta function $q(\boldsymbol{H}) = \delta(\boldsymbol{H}, \hat{\boldsymbol{H}})$ defined in (E.8). Substituting this delta function and the margin (E.12) in, the large margin training criterion (E.1) with the hinge loss criterion function (E.2) can be further written as minimising:

$$\mathcal{F}(\hat{\boldsymbol{H}}) = \mathrm{KL}\big(q(\boldsymbol{H}) || p(\boldsymbol{H})\big) + \sum_{n=1}^{N} \Bigg[\max_{W, \rho \neq W_n, \rho_n} \Big\{ \mathcal{L}(W, W_n) - \frac{1}{K} \sum_{k=1}^{K} \left(\hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n; \rho_n) - \hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W; \rho) \right) \Big\} \Bigg]_+ \tag{E.13}$$

where, for each training instance with index n, the best competing hypothesis and segmentation pair (W, ρ) is found over all possible hypotheses and segmentations[2] except the

[1] The discriminant function maps the input $\boldsymbol{O}$ directly to a class label W by choosing the class maximising this function.

[2] These possible hypotheses and segmentations can be obtained from a denominator lattice [147, 209].

reference with the corresponding segmentation (W_n, ρ_n), and ρ_n is the most likely segmentation obtained by the HMM as described in equation (5.25).

In the large margin training criterion (E.13), the Kullback-Leibler (KL) divergence is defined as $\mathrm{KL}\big(q(\boldsymbol{H})||p(\boldsymbol{H})\big) = \int q(\boldsymbol{H}) \log q(\boldsymbol{H}) d\boldsymbol{H} - \int q(\boldsymbol{H}) \log p(\boldsymbol{H}) d\boldsymbol{H}$. This criterion (E.13) can be written in a similar form to the large margin criterion described in (5.12) in section 5.2. $\int q(\boldsymbol{H}) \log q(\boldsymbol{H}) d\boldsymbol{H}$ is the negative of the Shannon entropy. Since $q(\boldsymbol{H})$ is a delta function, this entropy is an infinite but constant value. In the large margin training criterion (E.13), the best competing hypotheses are found given K parameter samples $\{\hat{\boldsymbol{\eta}}_{z_n^{(k)}}\}_{k=1}^K$. However, this leads to inefficiency in training. Rather than optimising criterion (E.13) directly, an upper bound can be used instead. By moving out the summation over k from the maximisation, the minimisation criterion (E.13) then can be relaxed to its upper bound. Thus the aim becomes to minimise the upper bound $\mathcal{F}_{\text{up}}(\hat{\boldsymbol{H}})$:

$$\mathcal{F}(\hat{\boldsymbol{H}}) \leq \mathcal{F}_{\text{up}}(\hat{\boldsymbol{H}}) = -\log p(\hat{\boldsymbol{H}}) + \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} \Bigg[\max_{W, \rho \neq W_n, \rho_n} \Big\{ \mathcal{L}(W, W_n) - \Big(\hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n; \rho_n) - \hat{\boldsymbol{\eta}}_{z_n^{(k)}}^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W; \rho) \Big) \Big\} \Bigg]_+ \tag{E.14}$$

Given the sampled indicators $\{\boldsymbol{z}^{(k)}\}_{k=1}^K$, the number of represented experts (which are the experts that have associated data) can be determined: $M = \big|\{\boldsymbol{z}^{(k)}\}_{k=1}^K\big|$. Let the prior distribution of $\boldsymbol{H}$ be $p(\boldsymbol{H}) = \prod_{m=1}^{\infty} p(\boldsymbol{\eta}_m)$. For the unrepresented experts (with index $m > M$), minimising criterion (E.14) yields the mode of the prior distribution $p(\boldsymbol{\eta}_m)$. For the parameters of the represented experts $\hat{\boldsymbol{H}}_r = \{\hat{\boldsymbol{\eta}}_m\}_{m=1}^M$, the large margin training criterion (E.14) can be reorganized as M minimisation criteria:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{H}}_r) = \sum_{m=1}^{M} \mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) \tag{E.15}$$

where:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) = -\log p(\hat{\boldsymbol{\eta}}_m) + \frac{1}{K} \sum_{k=1}^{K} \sum_{\forall z_n^{(k)} = m, \forall n} \Bigg[\max_{W, \rho \neq W_n, \rho_n} \Big\{ \mathcal{L}(W, W_n) - \Big(\hat{\boldsymbol{\eta}}_m^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W_n; \rho_n) - \hat{\boldsymbol{\eta}}_m^{\mathsf{T}} \Phi(\boldsymbol{O}_n, W; \rho) \Big) \Big\} \Bigg]_+ \tag{E.16}$$

Assume the prior of each expert's parameter is a Gaussian distribution $p(\boldsymbol{\eta}_m) = \mathcal{N}(\boldsymbol{\mu}_\eta, \Sigma_\eta)$ with mean $\boldsymbol{\mu}_\eta$ and a scaled identity matrix covariance $\Sigma_\eta = C\boldsymbol{I}$, then the mth criterion

can be further written as:

$$\mathcal{F}_{\text{LM}}(\hat{\boldsymbol{\eta}}_m) \propto \frac{1}{2}||\hat{\boldsymbol{\eta}}_m - \boldsymbol{\mu}_\eta||^2 + \frac{C}{K}\sum_{k=1}^{K}\sum_{\forall z_n^{(k)}=m,\forall n}\Big[\max_{W,\rho\neq W_n,\rho_n}\Big\{\hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\rho)+ \mathcal{L}(W,W_n)\Big\} - \hat{\boldsymbol{\eta}}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\rho_n)\Big]_+ \tag{E.17}$$

This criterion (E.17) is closely related to the large margin training criterion for each expert discussed in section 5.3.3, and this relationship will be discussed in detail in the following section.

E.2.3 The Relationship with Large Margin Training for Each Expert

Criterion (E.17) has the same form as the large margin training criterion for each expert described in section 5.3.3 (this type of model is denoted as the iLLM$_{\text{LM}^*}$). In the iLLM$_{\text{LM}^*}$, structured SVMs are trained with the data associated with each expert, and the training criterion for the mth expert in the kth iteration is described in (5.38):

$$\mathcal{F}_{\text{LM}}(\boldsymbol{\eta}_m) = \frac{1}{2}||\boldsymbol{\eta}_m - \boldsymbol{\mu}_\eta||^2 + C\sum_{\forall z_n^{(k)}=m,\forall n}\Big[\max_{W,\rho\neq W_n,\rho_n}\Big\{\boldsymbol{\eta}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W,\rho)+ \mathcal{L}(W,W_n)\Big\} - \boldsymbol{\eta}_m^{\mathsf{T}}\Phi(\boldsymbol{O}_n,W_n,\rho_n)\Big]_+ \tag{E.18}$$

Comparing criteria (E.17) and (E.18), the summation bounds are different. The bound of summation in (E.18) is only for the kth set of samples, whereas in criterion (E.17) the bound of summation is for all $k \in \{1,\ldots,K\}$. This means the training data are replicated K times in large margin training for all the experts (discussed in this appendix). For the nth training instance, it might be allocated to the same expert m in different iteration k. Thus, the most competing hypothesis W can be cached, and the cached hypothesis can be reused when this training instance is associated with the same expert again.

When the margin defined in (E.11) is approximated by one sample (namely $K = 1$), the large margin training criterion (E.17) becomes the same as criterion (E.18), which is the large margin training criterion for each expert of the iLLM$_{\text{LM}^*}$. Then the iterative process of optimising $\hat{q}(\boldsymbol{H})$ and $\hat{q}(\boldsymbol{\Theta},\boldsymbol{z})$ described in Algorithm 6 becomes the Gibbs sampling style training process described in section 5.3.3, which is the iterative process described in Algorithm 1 by replacing sampling for each expert with large margin training.

E.3 Classification

In section 5.3.1, Bayesian inference of the infinite log-linear model was discussed, where Gibbs sampling is used in training. For each draw, the number of represented experts (or components) is determined, and the mixture weight $\pi_m^{(k)}$ corresponding to each is proportional to the number of data associated with that expert. All the other model parameters $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{H}^{(k)}\}$ are also sampled. In classification, these mixture weights and other sampled model parameters are summed over to approximate the integral over all the model parameters, as described in (5.28). In this section, a different perspective on classification will be discussed, where the mixture weights $\boldsymbol{\pi}$ are considered as been marginalised out in the model as that in the Chinese restaurant process (CRP).

In large margin training of all experts for the infinite structured discriminative model, the infinite mixture of experts framework based on the CRP (as illustrated in Figure 4.10) is used. Indicator variables $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ corresponding to the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$ are introduced, and the mixture weights $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{\infty}$ are marginalised out (as in the CRP discussed in section 4.3.3). As described in (E.9), the optimal distribution of the model parameters is decomposed $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \approx \hat{q}(\boldsymbol{H})\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, where the optimal distribution for the experts is a Dirac delta function $\hat{q}(\boldsymbol{H}) = \delta(\boldsymbol{H}, \hat{\boldsymbol{H}})$, and samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}\}$ are drawn from distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$. Then, given the training data $\mathcal{D}$ and a new input $\boldsymbol{O}$, the conditional probability of the sentence W (corresponding to $\boldsymbol{O}$) can be described as:

$$
\begin{aligned}
P(W|\boldsymbol{O}, \mathcal{D}) &= \int \sum_{\boldsymbol{z}} P(W|\boldsymbol{O}, \boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) \hat{q}(\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{z}) d(\boldsymbol{\Theta}, \boldsymbol{H}) \\
&\approx \frac{1}{K} \sum_{k=1}^{K} P(W|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \hat{\boldsymbol{H}}, \boldsymbol{z}^{(k)}) \\
&= \frac{1}{K} \sum_{k=1}^{K} \sum_{z=1}^{\infty} P(W|\boldsymbol{O}, \hat{\boldsymbol{H}}, z) P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)})
\end{aligned} \tag{E.19}
$$

where z is the indicator variable corresponding to the new input $\boldsymbol{O}$, the samples $\{\boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}\}$ are drawn from distribution $\hat{q}(\boldsymbol{\Theta}, \boldsymbol{z})$, and $\hat{\boldsymbol{H}}$ are the parameters of $\hat{q}(\boldsymbol{H})$ which is a delta function defined in (E.8). The conditional distribution $P(W|\boldsymbol{O}, \hat{\boldsymbol{H}}, z)$ (for experts) is a log-linear model described in (5.24). $P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)})$ (for gating network) is the

component posterior distribution of the Gaussian mixture model similar to the component posterior distribution (5.21):

$$P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}) = \frac{P(z|\boldsymbol{z}^{(k)})\mathcal{N}\left(\varphi(\boldsymbol{O}); \boldsymbol{\theta}_z^{(k)}\right)}{\sum_z P(z|\boldsymbol{z}^{(k)})\mathcal{N}\left(\varphi(\boldsymbol{O}); \boldsymbol{\theta}_z^{(k)}\right)}, \quad z \in \{1, 2, \ldots, \infty\} \tag{E.20}$$

where $\mathcal{N}\left(\varphi(\boldsymbol{O}); \boldsymbol{\theta}_z^{(k)}\right)$ is the component likelihood given by a Gaussian distribution, and $\varphi(\boldsymbol{O})$ is a feature function, which maps the input $\boldsymbol{O}$ with various length to a feature with fixed dimension. The probability $P(z|\boldsymbol{z}^{(k)})$ is the mixture weights, which is given by the CRP described in (4.28). As discussed in section 5.3.1.1, only the represented experts are considered in classification. Given the sampled indicators $\boldsymbol{z}^{(k)}$, the number of the represented experts can be determined: $M_k = |\boldsymbol{z}^{(k)}|$, which is the number of the unique values in set $\boldsymbol{z}^{(k)}$. And each mixture weight can be described as $P(z|\boldsymbol{z}^{(k)}) \approx N_m^{(k)}/N$, where $N_m^{(k)}$ is the number of data associated with the mth expert, and N is the total number of training data. Let $\pi_z^{(k)} = P(z|\boldsymbol{z}^{(k)})$ and $\boldsymbol{\pi}^{(k)} = \{\pi_1^{(k)}, \ldots, \pi_{M_k}^{(k)}\}$, the component posterior distribution in (E.20) becomes the same form as the component posterior distribution in (5.21):

$$P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}) = P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{\pi}^{(k)}) \approx \frac{\pi_z^{(k)}\mathcal{N}\left(\varphi(\boldsymbol{O}); \boldsymbol{\theta}_z^{(k)}\right)}{\sum_z \pi_z^{(k)}\mathcal{N}\left(\varphi(\boldsymbol{O}); \boldsymbol{\theta}_z^{(k)}\right)}, \quad z \in \{1, \ldots, M_k\} \tag{E.21}$$

Given the number of represented expert M_k, the class posterior distribution (E.19) can be further written as:

$$P(W|\boldsymbol{O}, \mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} P(W|\boldsymbol{O}, \hat{\boldsymbol{H}}, z)P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{z}^{(k)}) \tag{E.22}$$

It is interesting to compare this class posterior distribution (E.22) with the class posterior distribution for the infinite structured discriminative model described in (5.28):

$$P(W, |\boldsymbol{O}, \mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{z=1}^{M_k} P(W|\boldsymbol{O}, \boldsymbol{H}^{(k)}, z)P(z|\boldsymbol{O}, \boldsymbol{\Theta}^{(k)}, \boldsymbol{\pi}^{(k)}) \tag{E.23}$$

They have the similar form, but the parameters of the experts have different meanings. In the class posterior distribution (E.23), the parameters of the experts $\boldsymbol{H}^{(k)} = \{\boldsymbol{\eta}_1^{(k)}, \ldots, \boldsymbol{\eta}_{M_k}^{(k)}\}$ are sampled from the conditional posterior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{z}^{(k)}, \mathcal{D})$ described in (5.33),

and $\boldsymbol{H}^{(k)}$ all vary with k. In contrast, the parameters $\hat{\boldsymbol{H}}$ in the class posterior distribution (E.22) are the parameters of $\hat{q}(\boldsymbol{H})$ described in (E.8). The parameters $\hat{\boldsymbol{H}}$ are estimated according to the large margin training criterion described in (E.1) with criterion function (E.2), and $\hat{\boldsymbol{H}}$ do not vary with k.

Appendix F

Structured Infinite Discriminative Models

In chapter 3.5.2, infinite structured discriminative models were discussed. In this type of model, the indicator variable corresponding to each utterance is a scalar, where the inputs to gating network are utterances and all the segments in an utterance share the same indicator. This might limit the flexibility of the gating network. In order to make better use of the data, a more granular (vector) indicator will be introduced. By doing so, different sub-sentence units (such as words or phones) can be associated with different experts, and more precise predictions can be made possible. This type of model is called the *structured infinite discriminative model*, which will be briefly discussed in this chapter.

F.1 An Equivalent Form of the Structured Discriminative Model

As discussed in the previous chapter, in Bayesian inference of the infinite structured discriminative models, given the sampled indicators variables $\boldsymbol{z}^{(k)} = \{z_1^{(k)}, \ldots, z_N^{(k)}\}$ corresponding to the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, the parameters of different experts are conditionally independent, hence the parameters $\boldsymbol{\eta}_m$ of each experts can be sampled separately as described in (5.33). Equivalently, the parameters of all the experts

also can be treated as a whole, and these parameters can be written in a stacked form[1]:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_M \end{bmatrix} \tag{F.1}$$

and for each expert the log-linear model described in (5.24) can be written in the following form:

$$P(W|\boldsymbol{O}, \boldsymbol{H}, z) \approx \frac{1}{\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, z)} \exp\left(\boldsymbol{H}^{\mathsf{T}} \Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z)\right) \tag{F.2}$$

where $\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, z)$ is a normalisation term, which can be obtained by marginalising over all passible word sequence W and segmentations $\boldsymbol{\rho}$ similar to (5.23):

$$\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, z) = \sum_{W \in \mathcal{W}} \sum_{\boldsymbol{\rho} \in \boldsymbol{P}_W} \exp\left(\boldsymbol{H}^{\mathsf{T}} \Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z)\right) \tag{F.3}$$

In the structured discriminative model (F.2), $\boldsymbol{\rho}$ is the most likely segmentation, e.g. the segmentation given by the HMM as described in (5.25). $\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z)$ is the extended joint feature:

$$\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z) = \begin{bmatrix} \delta(z, 1)\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \vdots \\ \delta(z, M)\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) \end{bmatrix} \tag{F.4}$$

where $\delta(\cdot)$ is the Kronecker delta, and $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$ is the joint feature discussed in section 3.5. Given the segmentation $\boldsymbol{\rho}$, the utterance and corresponding label can be described as $\boldsymbol{O} = \{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_{|\boldsymbol{\rho}|}\}$ and $W = \{w_1, \ldots, w_{|\boldsymbol{\rho}|}\}$, where $|\boldsymbol{\rho}|$ is the number of segments. Then the joint feature can be written as [210]:

$$\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) = \begin{bmatrix} \boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{|\boldsymbol{\rho}|} \delta(w_i, v_1)\varphi\big(\boldsymbol{O}_{(i)}\big) \\ \vdots \\ \sum_{i=1}^{|\boldsymbol{\rho}|} \delta(w_i, v_L)\varphi\big(\boldsymbol{O}_{(i)}\big) \\ \boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho}) \end{bmatrix} \tag{F.5}$$

where $\boldsymbol{\phi}_{\mathrm{ac}}(\cdot)$ denotes the acoustic features, and $\boldsymbol{\phi}_{\mathrm{lg}}(\cdot)$ denotes the language features. $\{v_1, \ldots, v_L\}$ denote all possible sub-sentence units (such as tri-phones) in the vocabulary. $\varphi\big(\boldsymbol{O}_{(i)}\big)$ are the features corresponding to the ith segment, which were discussed in detail in section 3.5.

[1] Since there are infinite number of parameters in an infinite model, M is infinite here. Similarly, in the following sections, without specification, M denotes a positive infinite integer.

It is worth noting that the structured discriminative model described in (F.2) is equivalent to the log-linear model described in (5.24), but with a different form of expression. For example, when $z = m$, in the extended joint feature only the elements associated with the m^{th} expert (the elements with $\delta(z, m)$) have non-zero values, all other elements are zero. Given the definition (F.1) of the parameters $\boldsymbol{H}$, then $\boldsymbol{H}^{\mathsf{T}}\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z)$ can be written as $\boldsymbol{\eta}_m^{\mathsf{T}}\Phi(\boldsymbol{O}, W, \boldsymbol{\rho})$. Therefore, the structured discriminative model described in (F.2) is equivalent to the log-linear model described in (5.24).

In Bayesian inference of infinite structured discriminative models (as described in section 5.3.1), given the sampled indicators $\boldsymbol{z}^{(k)} = \{z_1^{(k)}, \ldots, z_N^{(k)}\}$ corresponding to the training data $\mathcal{D} = \{(\boldsymbol{O}_1, W_1), \ldots, (\boldsymbol{O}_N, W_N)\}$, the number of represented experts M_k can be determined. The parameters for the represented experts can be written in the form of (F.1) with $M = M_k$, namely $\boldsymbol{H}_{\mathrm{r}} = \left[\boldsymbol{\eta}_1^{\mathsf{T}}, \ldots, \boldsymbol{\eta}_{M_k}^{\mathsf{T}}\right]^{\mathsf{T}}$. Then, sampling the parameters $\boldsymbol{\eta}_m$ of each represented expert described in (5.33) can be described as sampling the whole parameters $\boldsymbol{H}_{\mathrm{r}}$ of the represented experts according to:

$$p(\boldsymbol{H}_{\mathrm{r}}|\boldsymbol{z}^{(k)}, \mathcal{D}) \propto p(\boldsymbol{H}_{\mathrm{r}}) \prod_n P(W_n|\boldsymbol{O}_n, \boldsymbol{H}_{\mathrm{r}}, z_n^{(k)}) \tag{F.6}$$

where $p(\boldsymbol{H}_{\mathrm{r}})$ is the prior distribution, and $P(W_n|\boldsymbol{O}_n, \boldsymbol{H}_{\mathrm{r}}, z_n^{(k)})$ is the structured discriminative model for the $z_n^{(k)}$th expert as described in (F.2). The parameters $\boldsymbol{H}_{\mathrm{r}}^{(k)}$ can be sampled from (F.6) by using the Metropolis algorithm as discussed in section 5.3.1.5.

In conclusion, by using the form of the structured discriminative model described in (F.2) with the extended joint feature (F.4) and parameters (F.1), sampling the parameters of each experts in a separate fashion (5.33) is equivalent to sample the whole parameters according to (F.6).

F.1.1 Sharing of the Language Model

As discussed in section 3.5, in the structured discriminative model the features can be described as the form consisting of acoustic and language features, namely $\Phi(\boldsymbol{O}, W, \boldsymbol{\rho}) = \left[\boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}), \boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho})\right]^{\mathsf{T}}$ as described in (F.5). Then the extended joint feature (F.4) and

corresponding model parameters (F.1) can be described as follows:

$$\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z) = \begin{bmatrix} \delta(z,1)\boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \delta(z,1)\boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho}) \\ \vdots \\ \delta(z,M)\boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \delta(z,M)\boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho}) \end{bmatrix}, \qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{\eta}_1^{\mathrm{ac}} \\ \boldsymbol{\eta}_1^{\mathrm{lg}} \\ \vdots \\ \boldsymbol{\eta}_M^{\mathrm{ac}} \\ \boldsymbol{\eta}_M^{\mathrm{lg}} \end{bmatrix} \tag{F.7}$$

In discriminative models, it is important to tie the model parameters for robust parameter estimation. In the extended joint feature, the language model features $\boldsymbol{\phi}_{\mathrm{lg}}(\cdot)$ shared by different experts, and it is possible to tie these parameters $\{\boldsymbol{\eta}_1^{\mathrm{lg}}, \ldots, \boldsymbol{\eta}_M^{\mathrm{lg}}\}$ corresponding to the language model features. By using the extended joint feature $\Phi_{\mathrm{e}}(\cdot)$ as described in (F.7), tieing the language model parameters corresponding to different experts can be easily implemented. When the language model parameters $\{\boldsymbol{\eta}_1^{\mathrm{lg}}, \ldots, \boldsymbol{\eta}_M^{\mathrm{lg}}\}$ are tied, the extended joint feature and corresponding parameters can be described as:

$$\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, z) = \begin{bmatrix} \delta(z,1)\boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \vdots \\ \delta(z,M)\boldsymbol{\phi}_{\mathrm{ac}}(\boldsymbol{O}, W, \boldsymbol{\rho}) \\ \boldsymbol{\phi}_{\mathrm{lg}}(W, \boldsymbol{\rho}) \end{bmatrix}, \qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{\eta}_1^{\mathrm{ac}} \\ \vdots \\ \boldsymbol{\eta}_M^{\mathrm{ac}} \\ \boldsymbol{\eta}^{\mathrm{lg}} \end{bmatrix} \tag{F.8}$$

By sharing the language model parameters, in Bayesian inference of the infinite structured discriminative model, the parameters of the represented experts still can be sampled according to the posterior distribution described in (F.6).

F.1.2 Classification

As discussed in section 5.3.1.1, only the represented experts are used in classification in this work. Given the sampled indicators $\boldsymbol{z}^{(k)}$, the number of represented experts can be determined, namely $M_k = |\boldsymbol{z}^{(k)}|$. Then, the parameters $\boldsymbol{H}_{\mathrm{r}}$ of the represented experts can written in the form of (F.8) with $M = M_k$. In inference, parameters $\boldsymbol{H}_{\mathrm{r}}^{(k)}$ can be sampled according to the conditional posterior distribution (F.6). Given these sampled parameters $\boldsymbol{H}^{(k)}$, the parameters of different experts are known and can be written as follows:

$$\boldsymbol{\eta}_1^{(k)} = \begin{bmatrix} \boldsymbol{\eta}_1^{\mathrm{ac}(k)} \\ \boldsymbol{\eta}^{\mathrm{lg}(k)} \end{bmatrix}, \cdots, \boldsymbol{\eta}_M^{(k)} = \begin{bmatrix} \boldsymbol{\eta}_M^{\mathrm{ac}(k)} \\ \boldsymbol{\eta}^{\mathrm{lg}(k)} \end{bmatrix} \tag{F.9}$$

Then the class posterior distribution described in (5.28) can be used in classification. Alternatively, when using the form of the extended joint feature and corresponding model

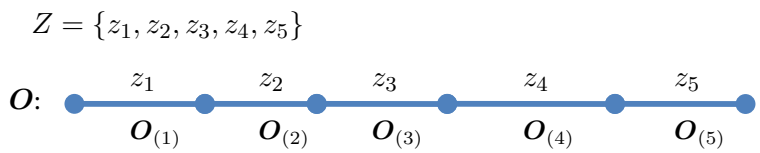


Figure F.1: *The vector indicator variable Z corresponding to an utterance.*

parameters (as described in (F.8)) directly, given an input utterance $\boldsymbol{O}$, the class label W can be found by maximising the following class posterior distribution, which has a similar form to (5.28):

$$P(W|\boldsymbol{O}, \mathcal{D}) \approx \frac{1}{K} \sum_{k=1}^{K} \sum_{z=1}^{M_k} P(W|\boldsymbol{O}, \boldsymbol{H}_{\mathrm{r}}^{(k)}, z) P(z|\boldsymbol{O}, \boldsymbol{\pi}^{(k)}, \boldsymbol{\Theta}^{(k)}) \tag{F.10}$$

where $P(W|\boldsymbol{O}, \boldsymbol{H}_{\mathrm{r}}^{(k)}, z)$ is the structured discriminative model described in (F.2), and $P(z|\boldsymbol{O}, \boldsymbol{\pi}^{(k)}, \boldsymbol{\Theta}^{(k)})$ is the probability given by the gating network, which is the same as that discussed in section 5.3.

F.2 Structured Infinite Discriminative Models

In the previous section, the parameters of the experts were treated as a whole and these parameters were described in the form of the extended joint feature. Share of the language model parameter among different experts was also discussed. So far the indicator variable z corresponding to an utterance is a scalar, whereas in this section a more general form of the indicator (a vector) will be discussed, and this type of vector indicator also can be incorporated into the extended joint feature described in (F.4). In order to distinguish from the scalar indicator z, the vector indictor corresponding to an utterance $\boldsymbol{O}$ is denoted as Z. Consider a segmented utterance $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \ldots, \boldsymbol{O}_{(|\boldsymbol{\rho}|)}\}$ with label $W = \{w_1, \ldots, w_{|\boldsymbol{\rho}|}\}$, the corresponding vector indicator can be described as $Z = \{z_1, \ldots, z_{|\boldsymbol{\rho}|}\}$, where $\boldsymbol{\rho}$ is the most likely segmentation and $|\boldsymbol{\rho}|$ is the number of segments. An example of the vector indicator with associated utterance is illustrated in Figure F.1. Given the vector indicator Z, the extended joint feature and corresponding model parameters described in (F.8) becomes

the form written as follows:

$$\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, Z) = \begin{bmatrix} \sum_{i=1}^{|\boldsymbol{\rho}|} \delta(z_i, 1)\boldsymbol{\phi}_{\mathrm{ac}}\big(\boldsymbol{O}_{(i)}, w_i, \rho_i\big) \\ \vdots \\ \sum_{i=1}^{|\boldsymbol{\rho}|} \delta(z_i, M)\boldsymbol{\phi}_{\mathrm{ac}}\big(\boldsymbol{O}_{(i)}, w_i, \rho_i\big) \\ \phi_{\mathrm{lg}}(W, \boldsymbol{\rho}) \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} \boldsymbol{\eta}_1^{\mathrm{ac}} \\ \vdots \\ \boldsymbol{\eta}_M^{\mathrm{ac}} \\ \boldsymbol{\eta}^{\mathrm{lg}} \end{bmatrix} \tag{F.11}$$

where $\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_{|\boldsymbol{\rho}|}\}$, $\boldsymbol{\phi}_{\mathrm{ac}}\big(\boldsymbol{O}_{(i)}, w_i, \rho_i\big)$ is the acoustic feature vector for a segment, and this feature vector is the utterance acoustic feature vector described in (F.5) with one segment in an utterance:

$$\boldsymbol{\phi}_{\mathrm{ac}}\big(\boldsymbol{O}_{(i)}, w_i, \rho_i\big) = \begin{bmatrix} \delta(w_i, v_1)\varphi\big(\boldsymbol{O}_{(i)}\big) \\ \vdots \\ \delta(w_i, v_L)\varphi\big(\boldsymbol{O}_{(i)}\big) \end{bmatrix} \tag{F.12}$$

where ρ_i is the segmentation corresponding to the ith segment, hence $|\rho_i| = 1$. $\{v_1, \ldots, v_L\}$ denote all possible sub-sentence units (such as tri-phones) in the vocabulary, and $\varphi\big(\boldsymbol{O}_{(i)}\big)$ are the features corresponding to the ith segment. By using vector indicators, the structured discriminative model described in (F.2) becomes:

$$P(W|\boldsymbol{O}, \boldsymbol{H}, Z) \approx \frac{1}{\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, Z')} \exp\Big(\boldsymbol{H}^{\mathsf{T}}\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, Z)\Big) \tag{F.13}$$

where $\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, Z')$ is the normalisation term:

$$\mathcal{Z}(\boldsymbol{H}, \boldsymbol{O}, Z') = \sum_{W \in \mathcal{W}} \sum_{\boldsymbol{\rho} \in \boldsymbol{P}_W} \exp\Big(\boldsymbol{H}^{\mathsf{T}}\Phi_{\mathrm{e}}(\boldsymbol{O}, W, \boldsymbol{\rho}, Z_{\boldsymbol{\rho}})\Big) \tag{F.14}$$

In calculating the normalisation term (F.14), all passible segmentations are considered. As an approximation, the denominator lattice can be used to provide all possible segmentations for an utterance. Since vector indicator variables are introduced, in an utterance different segments can be associated with different experts. Thus, the indicator variable set Z' corresponding to all these possible segments is introduced. Z' indicates the assignments of all the arcs (to different experts) in the denominator lattice. The vector indicator corresponding to segmentation $\boldsymbol{\rho}$ is denoted as $Z_{\boldsymbol{\rho}}$. It is worth noting that Z' is treated as an additional given indicator set, and Z is part of Z'.

When the indicator variable is a scalar, each input to the gating network is an utterance. When a vector indicator variable Z is introduced, each scalar indicator z_i corresponds to a segment. Then each input to the gating network is a segment, and different scalar indicators for an utterance can indicate to different experts. Similar to the infinite structured

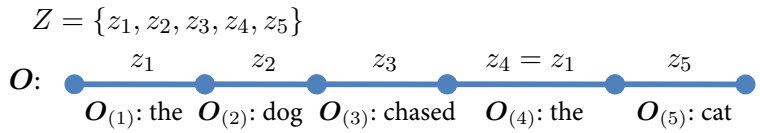


Figure F.2: *The indicators with constraints*

discriminative model where each utterance corresponds to a scalar indicator, when using a vector indicator for an utterance, the gating network can also be based on an infinite mixture model. In this work, the scalar indicators associated with the same utterance are assumed to be independent to each other. Thus, for an utterance $\boldsymbol{O} = \{\boldsymbol{O}_{(1)}, \ldots, \boldsymbol{O}_{(|\boldsymbol{\rho}|)}\}$ with vector indicator $Z = \{z_1, \ldots, z_{|\boldsymbol{\rho}|}\}$, the probability of the indicator Z given by the gating network can be decomposed:

$$P(Z|\boldsymbol{O}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \approx \prod_{i=1}^{|\boldsymbol{\rho}|} p(z_i|\boldsymbol{O}_{(i)}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \tag{F.15}$$

where $p(z_i|\boldsymbol{O}_{(i)}, \boldsymbol{\pi}, \boldsymbol{\Theta})$ is the component posterior distribution of the infinite mixture model as described in (5.21) with a segment input $\boldsymbol{O}_{(i)}$. By introducing the vector indicators, the conditional probability given by the infinite structured discriminative model described in (5.20) becomes:

$$\begin{aligned} P(W|\boldsymbol{O}, \mathcal{G}) &= \sum_{Z} P(W|\boldsymbol{O}, \boldsymbol{H}, Z) P(Z|\boldsymbol{O}, \boldsymbol{\pi}, \boldsymbol{\Theta}) \\ &\approx \sum_{Z} P(W|\boldsymbol{O}, \boldsymbol{H}, Z) \prod_{i=1}^{|\boldsymbol{\rho}|} P(z_i|\boldsymbol{O}_{(i)}, \boldsymbol{\pi}, \boldsymbol{\Theta}), \qquad Z \in \mathbb{Z} \end{aligned} \tag{F.16}$$

where $\mathcal{G} = \{\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{H}\}$ are all the parameters of the model, and $\mathbb{Z}$ is the set (with infinite size) that gives all possible values for vector Z. $P(W|\boldsymbol{O}, \boldsymbol{H}, Z)$ is the conditional probability of the structured discriminative model defined in (F.13). The model described in (F.16) is called the *structured infinite discriminative model* in this work.

F.2.1 Constraints on the Indicators

For the structured infinite discriminative model, in an utterance the segments having the same label can be associated with different experts. In an utterance, the segments having the same label have similar characteristics, it would be more appropriate to let these segments share the same model parameters. Then, constraints can be introduced to the indicators.

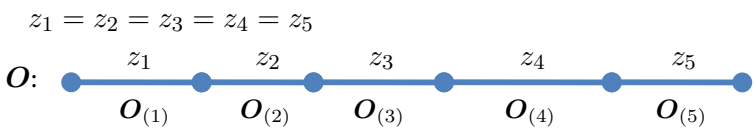


Figure F.3: *The segments in an utterance share the same indicator.*

One possible constraint is to let the segments with the same label in an utterance be associated with the same expert. An example is illustrated in Figure F.2. In this example, given that segments $\boldsymbol{O}_{(1)}$ and $\boldsymbol{O}_{(4)}$ have the same label, the associated indicators will be assigned the same value. By introducing constraints to the indicators, the structured infinite discriminative model then becomes the *constrained structured infinite discriminative model*.

F.2.1.1 Another Constraints

In the infinite structured discriminative model discussed in chapter 5, each utterance corresponds to a scalar utterance, namely different segments are associated with the same expert. Similarly, in the structured infinite discriminative model, the vector indicator (or all the scalar indicators) corresponding to an utterance can be constrained to share the same value. An example is illustrated in Figure F.3. It is worth noting that, in the structured infinite discriminative model, the inputs to the gating network are segments rather than utterances (in the infinite structured discriminative model).

F.3 Summary

In this appendix, the structured infinite discriminative models were briefly introduced. This type of model is a modification of the infinite structured discriminative model. In the infinite structured discriminative models scalar indicators are used for utterances, whereas the indicator variables are vectors in the structured infinite discriminative models. By introducing vector indicators, an infinite model (the gating network) can be built based on the segment inputs. This gives more flexibilities to the gating network, e.g. different segments in an utterance can be associated with different experts, and constraints can be added to different segments.

References

[1] A. Acero. "Acoustical and environmental robustness in automatic speech recognition". PhD thesis. Carnegie Mellon University, 1990.

[2] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. "HMM adaptation using vector Taylor series for noisy speech recognition". In: *Proceedings of ICSLP 2000*. Beijing, 2000, pp. 869–872.

[3] S. M. Ahadi and P. C. Woodland. "Combined Bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden Markov models". In: *Computer Speech and Language* 11.3, 1997, pp. 187–206.

[4] American Speech-Language-Hearing Association. *What Is Language? What Is Speech?* http://www.asha.org/public/speech/development/language_speech/. 2015.

[5] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. "A compact model for speaker-adaptive training". In: *Proceedings of International Conference on Spoken Language Processing (ICSLP)*. Vol. 2. ISCA. 1996, pp. 1137–1140.

[6] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. "An Introduction to MCMC for Machine Learning". In: *Machine Learning* 50.1-2, 2003, pp. 5–43.

[7] 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 ICASSP*. IEEE. Tokyo, 1986, pp. 49–52.

[8] 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". In: *Bulletin of the American Mathematicians Society* 73.3, 1967, pp. 360–363.

[9] L. E. Baum and G. R. Sell. "Growth transformations for functions on manifolds". In: *Pacific Journal of Mathematics* 27.2, 1968, pp. 211–227.

[10] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. "A maximum entropy approach to natural language processing". In: *Computational Linguistics* 22.1, 1996, pp. 39–71.

[11] J. M. Bernardo and A. F. Smith. *Bayesian theory*. John Wiley & Sons, 2009.

[12] J. Bertoin. *Lévy processes*. Cambridge university press, 1998.

[13] P. Billingsley. *Probability and Measure*. John Wiley & Sons, 1995.

[14] Ø. Birkenes. "A framework for speech recognition using logistic regression". PhD thesis. Norwegian University of Science and Technology, 2007.

[15] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford university press, 1995.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.

[17] C. M. Bishop and J. Lasserre. "Generative or Discriminative? Getting the Best of Both Worlds". In: *Bayesian Statistics* 8, 2007, pp. 3–24.

[18] D. Blackwell and J. B. MacQueen. "Ferguson distributions via Pólya urn schemes". In: *The Annals of Statistics* 1.2, 1973, pp. 353–355.

[19] D. M. Blei and M. I. Jordan. "Variational inference for Dirichlet process mixtures". In: *Bayesian Analysis* 1.1, 2006, pp. 121–144.

[20] D. M. Blei, M. I. Jordan, T. L. Griffiths, and J. B. Tenenbaum. "Hierarchical topic models and the nested Chinese restaurant process". In: *Advances in neural information processing systems* 16, 2004, p. 17.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent Dirichlet allocation". In: *Journal of Machine Learning Research* 3, 2003, pp. 993–1022.

[22] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM. 1992, pp. 144–152.

[23] H. Bourlard and N. Morgan. *Connectionist speech recognition: a hybrid approach*. Kluwer Academic Publishers, 1994.

[24] C. J. C. Burges. "A tutorial on support vector machines for pattern recognition". In: *Data Mining and Knowledge Discovery* 2.2, 1998, pp. 121–167.

[25] Cambridge University Press. *Cambridge Dictionaries Online*. http://dictionary.cambridge.org/. 2015.

[26] S. K. Card, T. P. Moran, and A. Newell. "The keystroke-level model for user performance time with interactive systems". In: *Communications of the ACM* 23.7, 1980, pp. 396–410.

[27] C. Chelba and A. Acero. "Adaptation of maximum entropy capitalizer: Little data can help a lot". In: *Computer Speech and Language* 20.4, 2006, pp. 382–399.

[28] T. Chen, J. Morris, and E. Martin. "Probability density estimation via an infinite Gaussian mixture model: application to statistical process monitoring". In: *Applied Statistics* 55.5, 2006, pp. 699–715.

[29] W. Chou, C.-H. Lee, and B.-H. Juang. "Minimum error rate training based on N-best string models". In: *Proceedings of ICASSP*. Vol. 2. IEEE. 1993, pp. 652–655.

[30] CMU. *The CMU Pronouncing Dictionary*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict?in=thesis. 2015.

[31] C. Cortes and V. Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3, 1995, pp. 273–297.

[32] D. R. Cox. "The Regression Analysis of Binary Sequences". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 20.2, 1958, pp. 215–242.

[33] K. Crammer and Y. Singer. "On the algorithmic implementation of multiclass kernel-based vector machines". In: *Journal of Machine Learning Research* 2, 2001, pp. 265–292.

[34] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[35] B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[36] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, et al. "Multilingual representations for low resource speech recognition and keyword search". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2015, pp. 259–266.

[37] R. C. van Dalen, J. Yang, and M. J. F. Gales. *Generative Kernels and Score-Spaces for Classification of Speech: Progress Report III*. Tech. rep. CUED/F-INFENG/TR699. Cambridge University Engineering Department, 2015.

[38] R. C. van Dalen, J. Yang, H. Wang, A. Ragni, C. Zhang, and M. J. F. Gales. "Structured Discriminative Models Using Deep Neural-Network Features". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2015.

[39] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society* 39.1, 1977, pp. 1–38.

[40] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, et al. "Recent advances in deep learning for speech research at Microsoft". In: *Proceedings of ICASSP*. IEEE. 2013, pp. 8604–8608.

[41] T. G. Dietterich and G. Bakiri. "Solving multiclass learning problems via error-correcting output codes". In: *Journal of Artificial Intelligence Research*, 1995, pp. 263–286.

[42] K.-B. Duan and S. S. Keerthi. "Which is the best multiclass SVM method? An empirical study". In: *Multiple Classifier Systems*. Springer, 2005, pp. 278–285.

[43] R. O. Duda, P. E. Hart, et al. “Pattern classification and scene analysis”. In: *J. Wiley and Sons*, 1973,

[44] T. S. Ferguson. “A Bayesian Analysis of Some Nonparametric Problems”. In: *The Annals of Statistics* 1.2, 1973, pp. 209–230.

[45] B. de Finetti. “Funzione caratteristica di un fenomeno aleatorio”. In: 1931,

[46] G. D. Forney Jr. “The Viterbi algorithm”. In: *Proceedings of the IEEE* 61.3, 1973, pp. 268–278.

[47] M. Forsberg. “Why is speech recognition difficult?” In: *Chalmers University of Technology*, 2003,

[48] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar. “Conditional random fields in speech, audio, and language processing”. In: *Proceedings of the IEEE* 101.5, 2013, pp. 1054–1075.

[49] E. B. Fox. “Bayesian Nonparametric Learning of Complex Dynamical Phenomena”. Ph.D. Thesis. Cambridge, MA, USA: Massachusetts Institute of Technology, 2009.

[50] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. “A Sticky HDP-HMM with Application to Speaker Diarization”. In: *Annals of Applied Statistics* 5.2A, 2011, pp. 1020–1056.

[51] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. “An HDP-HMM for systems with state persistence”. In: *Proceedings of International Conference on Machine Learning (ICML)*. ACM. 2008, pp. 312–319.

[52] S. Furui. “Speaker-independent isolated word recognition using dynamic features of speech spectrum”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 34.1, 1986, pp. 52–59.

[53] S. B. Gabriel et al. “The structure of haplotype blocks in the human genome”. In: *Science* 296.5576, 2002, pp. 2225–2229.

[54] M. J. F. Gales. “Discriminative models for speech recognition”. In: *Information Theory and Applications Workshop, 2007*. IEEE. 2007, pp. 170–176.

[55] M. J. F. Gales. "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition". In: *Computer Speech and Language* 12, 1998, pp. 75–98.

[56] M. J. F. Gales. "Model-based approaches to handling uncertainty". In: *Robust Speech Recognition of Uncertain or Missing Data*. Springer, 2011, pp. 101–125.

[57] M. J. F. Gales. "Model-based Techniques for Noise Robust Speech Recognition". PhD thesis. Cambridge, UK: University of Cambridge, 1995.

[58] M. J. F. Gales, B. Jia, X. Liu, K. C. Sim, P. C. Woodland, and K. Yu. "Development of the CUHTK 2004 RT04 Mandarin conversational telephone speech transcription system". In: *Proceedings of ICASSP*. IEEE. 2005, pp. 841–844.

[59] M. J. F. Gales, K. M. Knill, and A. Ragni. "UNICODE-BASED GRAPHEMIC SYSTEMS FOR LIMITED RESOURCE LANGUAGES". In: 2015,

[60] M. J. F. Gales and M. I. Layton. "SVMs, score-spaces and maximum margin statistical models". In: *Beyond HMM workshop, ATR*. 2004.

[61] M. J. F. Gales and S. J. Young. "Cepstral parameter compensation for HMM recognition in noise". In: *Speech communication* 12.3, 1993, pp. 231–239.

[62] M. J. F. Gales and S. J. Young. "Robust speech recognition in additive and convolutional noise using parallel model combination". In: *Computer Speech and Language* 9.4, 1995, pp. 289–307.

[63] M. J. F. Gales and F. Flego. "Discriminative classifiers with adaptive kernels for noise robust speech recognition". In: *Computer Speech and Language* 24.4, 2010, pp. 648–662.

[64] M. J. F. Gales, K. M. Knill, A. Ragni, and S. P. Rath. "Speech recognition and keyword spotting for low resource languages: Babel project research at CUED". In: *Spoken Language Technologies for Under-Resourced Languages*. 2014.

[65] M. J. F. Gales, S. Watanabe, and E. Fosler-Lussier. "Structured Discriminative Models For Speech Recognition". In: *IEEE Signal Processing Magazine*, 2012,

[66] M. J. F. Gales and S. J. Young. "The application of hidden Markov models in speech recognition". In: *Foundations and Trends in Signal Processing*. 2007, pp. 195–304.

[67] A. Ganapathiraju, J. E. Hamaker, and J. Picone. "Applications of support vector machines to speech recognition". In: *IEEE Transactions on Signal Processing* 52.8, 2004, pp. 2348–2355.

[68] J.-L. Gauvain and C.-H. Lee. "Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains". In: *IEEE Transactions on Speech and Audio Processing* 2.2, 1994, pp. 291–298.

[69] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall, 2014.

[70] S. J. Gershman and D. M. Blei. "A tutorial on Bayesian nonparametric models". In: *Journal of Mathematical Psychology* 56.1, 2011, pp. 1–12.

[71] J. Geweke. "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments". In: *Bayesian Statistics*. Oxford University Press, 1992, pp. 169–193.

[72] Z. Ghahramani and T. L. Griffiths. "Infinite latent feature models and the Indian buffet process". In: *Advances in neural information processing systems*. 2005, pp. 475–482.

[73] L. Gillick and S. J. Cox. "Some statistical issues in the comparison of speech recognition algorithms". In: *Proceedings of ICASSP*. IEEE. 1989, pp. 532–535.

[74] V. Goel and W. J. Byrne. "Minimum Bayes-risk automatic speech recognition". In: *Computer Speech and Language* 14.2, 2000, pp. 115–135.

[75] S. Goldwater, T. L. Griffiths, and M. Johnson. "Contextual dependencies in unsupervised word segmentation". In: *In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. 2006.

[76] 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 con-

tinuous speech recogniser on the ARPA noise spoke task". In: *ARPA Workshop on Spoken Language System Technology*. ARPA. 1995.

[77] D. Görür and C. E. Rasmussen. "Dirichlet process Gaussian mixture models: choice of the base distribution". In: *Journal of Computer Science and Technology* 25.4, 2010, pp. 615–626.

[78] T. L. Griffiths and Z. Ghahramani. "The indian buffet process: An introduction and review". In: *The Journal of Machine Learning Research* 12, 2011, pp. 1185–1224.

[79] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. "Hidden conditional random fields for phone classification." In: *Proceedings of Interspeech*. 2005, pp. 1117–1120.

[80] M. M. Hasan and Y. Matsumoto. "Japanese-Chinese Cross-Language Information Retrieval: An Interlingua Approach". In: *Computational Linguistics* 5.2, 2000, pp. 59–86.

[81] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. 2nd ed. Springer, 2009.

[82] E. Haugen. "The Curse of Babel". English. In: *Daedalus* 102.3, 1973, pp. 47–57. ISSN: 00115266. URL: http://www.jstor.org/stable/20024145.

[83] K. A. Heller and Z. Ghahramani. "A nonparametric Bayesian approach to modeling overlapping clusters". In: *International Conference on Artificial Intelligence and Statistics*. 2007, pp. 187–194.

[84] H. Hermansky, D. W. Ellis, and S. Sharma. "Tandem connectionist feature extraction for conventional HMM systems". In: *Proceedings of ICASSP*. Vol. 3. IEEE. 2000, pp. 1635–1638.

[85] G. Hinton, L. Deng, D. Yu, D. Dahl, A.-R. Mohamed, N. Jaitly, et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition". In: *IEEE Signal Processing Magazine*, 2012, pp. 2–17.

[86] N. L. Hjort. "Nonparametric Bayes estimators based on beta processes in models for life history data". In: *The Annals of Statistics*, 1990, pp. 1259–1294.

[87] P. D. Hoff. *Measure and probability*. 2013.

[88] C.-W. Hsu and C.-J. Lin. "A comparison of methods for multiclass support vector machines". In: *IEEE Transactions on Neural Networks* 13.2, 2002, pp. 415–425.

[89] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0130226165.

[90] T. Jaakkola, M. Meila, and T. Jebara. "Maximum Entropy Discrimination". In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1999, pp. 470–476.

[91] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. "Adaptive mixtures of local experts". In: *Neural Computation* 3.1, 1991, pp. 79–87.

[92] T. Jebara. "Discriminative, Generative and Imitative Learning". PhD thesis. Cambridge, MA, USA: Massachusetts Institute of Technology, 2001.

[93] T. Jebara. "Multitask sparsity via maximum entropy discrimination". In: *Journal of Machine Learning Research* 12, 2011, pp. 75–110.

[94] H. Jiang, X. Li, and C. Liu. "Large margin hidden Markov models for speech recognition". In: *IEEE Transactions on Audio, Speech and Language Processing* 14.5, 2006, pp. 1584–1595.

[95] Q. Jiang, J. Zhu, M. Sun, and E. P. Xing. "Monte Carlo methods for maximum margin supervised topic models". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1592–1600.

[96] T. Joachims, T. Finley, and C.-N. J. Yu. "Cutting-plane training of structural SVMs". In: *Machine Learning* 77.1, 2009, pp. 27–59.

[97] M. I. Jordan. "Hierarchical models, nested models and completely random measures". In: *Frontiers of Statistical Decision Making and Bayesian Analysis*, 2010,

[98] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2, 1999, pp. 183–233.

[99] M. I. Jordan and R. A. Jacobs. "Hierarchical mixtures of experts and the EM algorithm". In: *Neural Computation* 6.2, 1994, pp. 181–214.

[100] B.-H. Juang, W. Hou, and C.-H. Lee. "Minimum classification error rate methods for speech recognition". In: *IEEE Transactions on Speech and Audio Processing* 5.3, 1997, pp. 257–265.

[101] S. M. Katz. "Estimation of probabilities from sparse data for the language model component of a speech recognizer". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 35.3, 1987, pp. 400–401.

[102] J. Kingman. "Completely random measures". In: *Pacific Journal of Mathematics* 21.1, 1967, pp. 59–78.

[103] S. Kumar and W. Byrne. *Minimum Bayes-risk decoding for statistical machine translation*. Tech. rep. DTIC Document, 2004.

[104] J. Lafferty, A. McCallum, and F. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of International Conference on Machine Learning (ICML)*. ACM. San Francisco, CA, USA, 2001, pp. 282–289.

[105] J. Lasserre. "Hybrid of Generative and Discriminative Methods for Machine Learning". PhD thesis. University of Cambridge, 2008.

[106] M. I. Layton and M. J. F. Gales. "Augmented statistical models for speech recognition". In: *Proceedings of ICASSP*. 2006.

[107] M. I. Layton and M. J. F. Gales. *Maximum Margin Training of Generative Kernels*. Tech. rep. CUED/F-INFENG/TR.484. Cambridge University Engineering Department, 2004.

[108] M. I. Layton and M. J. F. Gales. *SVMs, score-spaces and maximum margin statistical models*. Tech. rep. CUED/F-INFENG/TR.484. Cambridge University Engineering Department, 2004.

[109] M. I. Layton. "Augmented statistical models for classifying sequence data". PhD thesis. University of Cambridge, 2006.

[110] C. J. Leggetter and P. C. Woodland. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models". In: *Computer Speech and Language* 9.2, 1995, pp. 171–185.

[111] R. G. Leonard. "A database for speaker-independent digit recognition". In: *ICASSP*. Vol. 9. IEEE. 1984, pp. 328–331.

[112] V. I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet Physics - Doklady* 10.8, 1966, pp. 707–710.

[113] J. Li, S. M. Siniscalchi, and C.-H. Lee. "Approximate test risk minimization through soft margin estimation". In: *Proceedings of ICASSP*. Vol. 4. IEEE. 2007, pp. 653–656.

[114] S. Li and C.-R. Huang. "Word Boundary Decision with CRF for Chinese Word Segmentation." In: *PACLIC*. 2009, pp. 726–732.

[115] H. Liao and M. J. F. Gales. "Adaptive training with joint uncertainty decoding for robust recognition of noisy data". In: *ICASSP*. Vol. 4. IEEE. 2007, pp. IV–389.

[116] X. Liu, M. J. F. Gales, and P. C. Woodland. "Automatic complexity control for HLDA systems". In: *Proceedings of ICASSP*. Hong Kong, 2003.

[117] J. Lööf, R. Schlüter, and H. Ney. "Discriminative adaptation for log-linear acoustic models." In: *Proceedings of Interspeech*. 2010, pp. 1648–1651.

[118] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney. "Investigations on Error Minimizing Training Criteria for Discriminative Training in Automatic Speech Recognition". In: *Proceedings of Interspeech*. ISCA. 2005, pp. 2133–2136.

[119] R. Malouf. "A comparison of algorithms for maximum entropy parameter estimation". In: *Proceedings of the 6th Conference on Natural Language Learning*. Vol. 20. Association for Computational Linguistics. 2002, pp. 1–7.

[120] L. Mangu, E. Brill, and A. Stolcke. "Finding consensus among words: lattice-based word error minimization." In: *Proceedings of Eurospeech*. 1999.

[121] A. McCallum, D. Freitag, and F. C. N. Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation." In: *Proceedings of International Conference on Machine Learning (ICML)*. Vol. 17. 2000, pp. 591–598.

[122] Q. McNemar. "Note on the sampling error of the difference between correlated proportions or percentages". In: *Psychometrika* 12.2, 1947, pp. 153–157.

[123] D. Meko. *Applied Time Series Analysis*. 2011.

[124] G. Mendels, E. Cooper, V. Soto, J. Hirschberg, M. J. F. Gales, K. M. Knill, et al. "Improving speech recognition and keyword search for low resource languages using web data". In: *Proceedings of Interspeech*. 2015.

[125] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6, 1953, pp. 1087–1092.

[126] N. Metropolis and S. Ulam. "The monte carlo method". In: *Journal of the American statistical association* 44.247, 1949, pp. 335–341.

[127] B. T. Meyer, M. Wächter, T. Brand, and B. Kollmeier. "Phoneme confusions in human and automatic speech recognition." In: *Proceedings of Interspeech*. 2007, pp. 1485–1488.

[128] P. J. Moreno, B. Raj, and R. M. Stern. "A vector Taylor series approach for environment-independent speech recognition". In: *Proceedings of ICASSP*. Vol. 2. IEEE. 1996, pp. 733–736.

[129] Moses. *Genesis 11:1–9*. https://www.biblegateway.com/passage/?search=Genesis+11&version=NIV. 2011.

[130] A. Nádas. "A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 31.4, 1983, pp. 814–817.

[131] R. M. Neal. "Markov chain sampling methods for Dirichlet process mixture models". In: *Journal of Computational and Graphical Statistics* 9.2, 2000, pp. 249–265.

[132] A. Y. Ng and M. I. Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in neural information processing systems* 14, 2002, p. 841.

[133] P. Orbanz and Y. W. Teh. "Bayesian Nonparametric Models". In: *Encyclopedia of Machine Learning*. Springer, 2010.

[134] Oxford University Press. *Oxford Advanced Learner's Dictionary*. http://www.oxfordlearnersdictionaries.com/. 2015.

[135] J. Paisley and L. Carin. "Nonparametric factor analysis with beta process priors". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 777–784.

[136] K. A. Papineni. "Discriminative training via linear programming". In: *Proceedings of ICASSP*. Vol. 2. IEEE. 1999, pp. 561–564.

[137] D. Pearce and H.-G. Hirsch. "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions". In: *Proceedings of International Conference on Spoken Language Processing (ICSLP)*. ISCA. 2000, pp. 29–32.

[138] J. Pitman. *Combinatorial Stochastic Processes*. Berlin: Springer-Verlag, 2006.

[139] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. "Large Margin DAGs for Multiclass Classification." In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 12. 1999, pp. 547–553.

[140] D. Povey and P. C. Woodland. "Minimum phone error and I-smoothing for improved discriminative training". In: *Proceedings of ICASSP*. IEEE. Orlando, 2002, pp. 105–108.

[141] D. Povey. "Discriminative Training for Large Vocabulary Speech Recognition". PhD thesis. University of Cambridge, 2003.

[142] D. Povey, A. Ghoshal, et al. "fhThe Kaldi speech recognition toolkit". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2011.

[143] 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 ICASSP*. IEEE. 2008, pp. 4057–4060.

[144] J. W. Pratt. "F. Y. Edgeworth and R. A. Fisher on the Efficiency of Maximum Likelihood Estimation". In: *The Annals of Statistics* 4.3, 1976, pp. 501–514.

[145] A. Quattoni, M. Collins, and T. Darrell. "Conditional random fields for object recognition". In: *Advances in Neural Information Processing Systems (NIPS)*. 2004, pp. 1097–1104.

[146] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, 1989, pp. 257–286.

[147] A. Ragni. "Discriminative models for Speech Recognition". PhD thesis. University of Cambridge, 2013.

[148] A. Ragni and M. J. F. Gales. "Derivative kernels for noise robust ASR". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2011, pp. 119–124.

[149] C. E. Rasmussen. "The Infinite Gaussian Mixture Model". In: *Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 554–560.

[150] C. E. Rasmussen and Z. Ghahramani. "Infinite Mixtures of Gaussian Process Experts". In: *NIPS*. 2001, pp. 881–888.

[151] M. Riedmiller and H. Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm". In: *IEEE International Conference on Neural Networks*. IEEE. 1993, pp. 586–591.

[152] J. Salomon, S. King, and M. Osborne. "Framewise phone classification using support vector machines". In: *Proceedings of International Conference on Spoken Language Processing (ICSLP)*. ISCA. 2002, pp. 2645–2648.

[153] G. Saon and D. Povey. "Penalty function maximization for large margin HMM training." In: *Proceedings of Interspeech*. 2008, pp. 920–923.

[154] S. Sarawagi and W. W. Cohen. "Semi-Markov conditional random fields for information extraction". In: *Advances in Neural Information Processing Systems (NIPS)*. 2004, pp. 1185–1192.

[155] K. Sato. *Lévy processes and infinitely divisible distributions*. Cambridge university press, 1999.

[156] L. J. Savage. "On Rereading R. A. Fisher". In: *The Annals of Statistics* 4.3, 1976, pp. 441–500.

[157] R. Schlüter and W. Macherey. "Comparison of discriminative training criteria". In: *Proceedings of ICASSP*. Vol. 1. IEEE. 1998, pp. 493–496.

[158] F. Seide, G. Li, and D. Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks". In: *Proceedings of Interspeech*. ISCA. 2011, pp. 437–440.

[159] M. L. Seltzer, D. Yu, and Y. Wang. "An investigation of deep neural networks for noise robust speech recognition". In: *Proceedings of ICASSP*. IEEE. 2013, pp. 7398–7402.

[160] J. Sethuraman. "A constructive definition of Dirichlet priors". In: *Statistica Sinica* 4, 1994, pp. 639–650.

[161] F. Sha and L. K. Saul. "Large margin Gaussian mixture modeling for phonetic classification and recognition". In: *Proceedings of ICASSP*. Vol. 1. IEEE. 2006, pp. 265–268.

[162] P.-Y. Shih, J.-F. Wang, H.-P. Lee, H.-J. Kai, H.-T. Kao, and Y.-N. Lin. "Acoustic and phoneme modeling based on confusion matrix for ubiquitous mixed-language speech recognition". In: *Proceedings of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. IEEE Computer Society. 2008, pp. 500–506.

[163] K. Shinoda and C.-H. Lee. "Structural MAP speaker adaptation using hierarchical priors". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 1997, pp. 381–388.

[164] N. Smith and M. J. F. Gales. "Speech recognition using SVMs". In: *Advances in neural information processing systems*. 2001, pp. 1197–1204.

[165] A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans. *Advances in Large Margin Classifiers*. MIT press, 2000.

[166] E. Snelson and Z. Ghahramani. "Compact approximations to Bayesian predictive distributions". In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 840–847.

[167] M. Stephens, N. J. Smith, and P. Donnelly. "A new statistical method for haplotype reconstruction from population data". In: *The American Journal of Human Genetics* 68.4, 2001, pp. 978–989.

[168] E. B. Sudderth. "Graphical Models for Visual Object Recognition and Tracking". PhD thesis. Cambridge, MA, USA: Massachusetts Institute of Technology, 2006.

[169] 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 ICASSP*. IEEE. 2008, pp. 4293–4296.

[170] 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 (ASRU)*. IEEE. 2007, pp. 347–352.

[171] K. Tanabe. "Penalized logistic regression machines: New methods for statistical prediction 1". In: *Proc. IBIS, Tokyo*, 2001, pp. 71–76.

[172] M. A. Tanner and W. H. Wong. "The calculation of posterior distributions by data augmentation". In: *Journal of the American statistical Association* 82.398, 1987, pp. 528–540.

[173] B. Taskar. "Learning Structured Prediction Models: A Large Margin Approach". PhD thesis. Stanford University, 2004.

[174] Y. W. Teh and M. I. Jordan. "Hierarchical Bayesian Nonparametric Models with Applications". In: *Bayesian Nonparametrics: Principles and Practice*. Ed. by N. Hjort, C. Holmes, P. Müller, and S. Walker. Cambridge University Press, 2010.

[175] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. "Hierarchical Dirichlet Processes". In: *Journal of the American Statistical Association* 101.476, 2006, pp. 1566–1581.

[176] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. "Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes". In: *Advances in Neural Information Processing Systems*. Vol. 17. 2005.

[177] Y. W. Teh. "Dirichlet Processes". In: *Encyclopedia of Machine Learning*. Springer, 2010.

[178] R. Thibaux and M. I. Jordan. "Hierarchical beta processes and the Indian buffet process". In: *International conference on artificial intelligence and statistics*. 2007, pp. 564–571.

[179] H. Thompson. "Best-first enumeration of paths through a lattice–an active chart parsing solution". In: *Computer Speech and Language* 4.3, 1990, pp. 263 –274.

[180] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. "Support vector machine learning for interdependent and structured output spaces". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 104.

[181] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. "Large margin methods for structured and interdependent output variables". In: *Journal of Machine Learning Research* 16, 2005, pp. 1453–1484.

[182] Z. Tuske, D. Nolden, R. Schluter, and H. Ney. "Multilingual MRASTA features for low-resource keyword search and speech recognition systems". In: *Proceedings of ICASSP*. IEEE. 2014, pp. 7854–7858.

[183] V. N. Vapnik. *Statistical learning theory*. Vol. 1. Wiley New York, 1998.

[184] S. R. S. Varadhan. *Probability Theory*. 2000.

[185] V. Venkataramani, S. Chakrabartty, and W. Byrne. "Support vector machines for segmental minimum Bayes risk decoding of continuous speech". In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2003, pp. 13–18.

[186] M. Vihola. "Dissimilarity measures for hidden Markov models and their application in multilingual speech recognition". MA thesis. Tampere University of Technology, 2001.

[187] A. J. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2, 1967, pp. 260–269.

[188] M. J. Wainwright and M. I. Jordan. "Graphical models, exponential families, and variational inference". In: *Foundations and Trends® in Machine Learning* 1.1-2, 2008, pp. 1–305.

[189] S. H. Walker and D. B. Duncan. "Estimation of the probability of an event as a function of several independent variables". In: *Biometrika* 54.1-2, 1967, pp. 167–179.

[190] B. Walsh. *Markov chain Monte Carlo and Gibbs sampling*. 2004.

[191] H. Wang, A. Ragni, M. J. F. Gales, K. M. Knill, P. C. Woodland, and C. Zhang. "Joint Decoding of Tandem and Hybrid Systems for Improved Keyword Spotting on Low Resource Languages". In: *Proceedings of Interspeech*. ISCA. 2015.

[192] Y. Wang. "Model-based Approaches to Robust Speech Recognition in Diverse Environments". PhD thesis. University of Cambridge, 2015.

[193] Y. Wang and M. J. F. Gales. "Speaker and noise factorization for robust speech recognition". In: *IEEE Transactions on Audio, Speech and Language Processing* 20.7, 2012, pp. 2149–2158.

[194] Wikipedia. *Statistical significance*. https://en.wikipedia.org/wiki/Statistical_significance.

[195] C. Wooters and M. Huijbregts. "The ICSI RT07s speaker diarization system". In: *Multimodal Technologies for Perception of Humans*. Springer, 2008, pp. 509–519.

[196] J. Yang, R. C. van Dalen, and M. J. F. Gales. "Infinite Support Vector Machines in Speech Recognition". In: *Proceedings of Interspeech*. 2013.

[197] J. Yang, R. C. van Dalen, S.-X. Zhang, and M. J. F. Gales. "Infinite Structured Support Vector Machines for Speech Recognition". In: *Proceedings of ICASSP*. 2014.

[198] J. Yang, A. Ragni, M. J. F. Gales, and K. M. Knill. "Log-linear System Combination Using Structured Support Vector Machines". In: *Proceedings of Interspeech*. 2016.

[199] J. Yang, C. Zhang, A. Ragni, M. J. F. Gales, and P. C. Woodland. "System Combination with Log-linear Models". In: *Proceedings of ICASSP*. IEEE. 2016.

[200] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain., D. Kershaw, X. Liu, et al. *The HTK book (for HTK version 3.4)*. Cambridge, UK: Cambridge University Engineering Department, 2006.

[201] S. J. Young and G. Bloothooft. *Corpus-Based Methods in Language and Speech Processing*. Vol. 2. Kluwer Academic Publishers, 1997.

[202] S. J. Young, J. J. Odell, and P. C. Woodland. "Tree-based state tying for high accuracy acoustic modelling". In: *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics. 1994, pp. 307–312.

[203] K. Yu. "Adaptive Training for Large Vocabulary Continuous Speech Recognition". PhD thesis. Cambridge, UK: University of Cambridge, 2006.

[204] A. L. Yuille and A. Rangarajan. "The concave-convex procedure". In: *Neural Computation* 15.4, 2003, pp. 915–936.

[205] A. Zellner. "Optimal Information Processing and Bayes's Theorem". English. In: *The American Statistician* 42.4, 1988, pp. 278–280. ISSN: 00031305.

[206] C. Zhai and J. Lafferty. "Model-based Feedback in the Language Modeling Approach to Information Retrieval". In: *Proceedings of the Tenth International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2001, pp. 403–410.

[207] A. Zhang, J. Zhu, and B. Zhang. "Max-margin infinite hidden Markov models". In: *Proceedings of International Conference on Machine Learning (ICML)*. 2014, pp. 315–323.

[208] S.-X. Zhang, A. Ragni, and M. J. F. Gales. "Structured Log Linear Models for Noise Robust Speech Recognition". In: *IEEE Signal Processing Letters* 17, 2010, pp. 945–948.

[209] S.-X. Zhang. "Structured Support Vector Machines for Speech Recognition". PhD thesis. University of Cambridge, 2014.

[210] S.-X. Zhang and M. J. F. Gales. "Structured SVMs for Automatic Speech Recognition". In: *IEEE Transactions on Audio, Speech and Language Processing* 21, 3 2013, pp. 544–555.

[211] J. Zheng and A. Stolcke. "Improved Discriminative Training Using Phone Lattices". In: *Proceedings of Interspeech*. ISCA. 2005, pp. 2125–2128.

[212] J. Zhu, N. Chen, H. Perkins, and B. Zhang. "Gibbs max-margin topic models with data augmentation". In: *Journal of Machine Learning Research* 15, 2014, pp. 1073–1110.

[213] J. Zhu, N. Chen, and E. Xing. "Infinite SVM: a Dirichlet Process Mixture of Large-margin Kernel Machines". In: *Proceedings of International Conference on Machine Learning (ICML)*. Bellevue, Washington, USA: ACM, 2011, pp. 617–624.

[214] J. Zhu, N. Chen, and E. P. Xing. "Bayesian inference with posterior regularization and applications to infinite latent SVMs". In: *Journal of Machine Learning Research* 15, 2014, pp. 1799–1847.

[215] 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 (ASRU)*. IEEE. 2009, pp. 152–157.