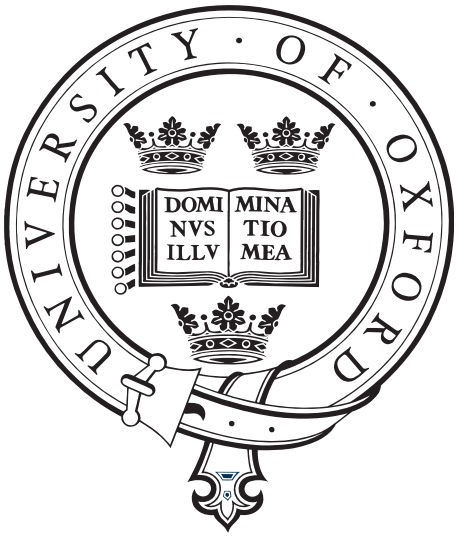
Classification of Partially Labelled Data using "Mixture of Expert" Models

Tjun Kiat Teo

Linacre College

University of Oxford



A thesis submitted for the degree of Doctor of Philosophy in Statistics

Hilary Term 2017

Acknowledgements

First and foremost, I would like to thank my wonderful supervisor, Professor Brain D. Ripley, for his excellent supervision. I would also like to thank my family, colleagues and friends for all their invaluable help and support along the way.

Abstract

In this thesis, we are concerned with the classification of partially labeled data. By partially labeled data we mean data where measurements are available from experimental units which are known to belong to one of a set of known classes but whose individual membership to subclasses within the known class is not known. Examples of these applications include fisheries research where fish lengths are available but sexual identities are not, sedimentology where information is available about the grain size distribution of a sample of sand but not its mineral composition and medical diagnosis where the symptoms of the patients are known but not disease classifications.

A popular way of handling such partially labelled data is to use a mixture of Gaussian densities. Relying on the assumption of Gaussian densities (more information), results in an more efficient (less variance) discrimination procedure if the modelling assumptions are satisfied. However in practice, these assumptions rarely hold and often some of the features are qualitative variables, and hence it is generally of the view that logistic discrimination is a more robust bet as it relies on fewer assumptions. We chose to use a mixture of logistic regressions, embedded within a hierarchical mixture of experts to classify our data.

We compared its use in the plug-in-approach (frequentist approach) versus the predictive approach (Bayesian approach) in classification. The density of parameters required for the predictive approach was obtained using Markov Chain Monte Carlo simulation.

Declaration

I declare that this thesis is wholly my own work unless otherwise stated. No part of this thesis has been accepted or is currently being submitted for any degree or diploma or certificate or other qualification in this university or elsewhere.

Candidate: Tjun Kiat Teo

Signed: _____ Date: 21 Apr 2017

Contents

Chapter 1

Introduction

1.1 Overview and History

In this thesis, we investigate the classification of partially labeled data with our chosen statistical model. By partially labelled data we mean data where measurements are available from experimental units which are known to belong to one of a set of known classes but whose individual membership to subclasses within the known class is not "known". Examples of these applications include fisheries research where fish lengths are available but sexual identities are not, sedimentology where information is available about the grain size distribution of a sample of sand but not its mineral composition and medical diagnosis where the symptoms of the patients are known but not disease classifications. [Titterington et al., 1985, pp ix].

Mixture models are the most popular way of handling such data. However

as explicit estimators do not exist for the estimators of mixture models, necessitating the use of numerical methods, usage of mixture models were not widespread till computers were invented and the publication of the seminal paper of Dempster et al. [1977] on the EM algorithm which greatly simplified fitting of mixture models by maximum likelihood.

With the advent of inexpensive, high speed computers in conjunction with rapid development in posterior simulation techniques such as Markov Chain Monte Carlo (MCMC) methods, usage of Bayesian methods in mixtures have also taken off and this will be one of the main themes of this thesis.

1.2 Statistical Decision Theory

1.2.1 Classification

In this thesis, we are interested in classification. The framework for classification is as follows: certain objects are classified as coming from one of fixed number of class say $1, \ldots, C$. Each object gives rise to certain measurements which together form the *feature vector* X. The proportion π^c of class c cases in the population under examination is some known or unknown π^c. Feature vectors from class c are distributed according to some density $f^c(x)$. The task at hand is to classify an object to one of the C classes on the observed value of $X = x$. And this is done by calculating the conditional posterior probability $f(c|x)$ for all possible classes and choosing the class that gives the maximum posterior probability.

1.2.2 Diagnostic verses Sampling Paradigm

A popular way to classify such data is to use a mixture of normal distributions which is an example of the sampling paradigm [Dawid, 1976] in statistical pattern recognition. In the sampling paradigm, where the models are usually termed generative methods, a parametric or non-parametric model is formed for the distribution of features for examples from each class and statistical decision theory is used to find an optimal classification. In the diagnostic paradigm, we are not keen on what the classes looked like but only given an example in what the distribution over classes is *for similar examples*. The main method for this approach came to be known as *logistic discrimination*. Both assume a parametric model of the joint density $f(x, c; \theta)$. In the sampling paradigm we are interested in $f(x, c; \theta) = \pi^c f^c(x; \theta)$. In the diagnostic paradigm, we estimate the posterior probabilities $f(c|x, \theta)$ directly via $f(x, c; \theta) = f(c|x, \theta) f(x; \theta)$ where any information about θ in the unconditional density $p(x; \theta)$ is normally discarded by conditioning on the observed $x's$. In the sampling paradigm, the posterior probability has to be calculating using Bayes rule where

$$f(c|x) = \frac{\pi^c f^c(x)}{\sum_{r=1}^{C} \pi^r f^r(x)}. \tag{1.2.1}$$

Each of these approaches have it strengths and weaknesses. The sampling paradigm, by making some modelling assumptions about the density

of $f(x)$ results in a more efficient (less variance) discrimination procedure. Logistic discrimination, by modelling the posterior probabilities directly, is considered less sensitive to modelling assumptions but is unable to make use of unlabelled data which the sampling paradigm is able to.

1.3 Plug in Approach verses Predictive Approach

Evers [2007] developed a general purpose implementation of mixture of experts models for public use in R. His thesis however only covered the frequentist fitting of mixture of experts. In this thesis, we attempted to compare frequentist fitting (plug-in-approach) with the Bayesian fitting (predictive approach) of mixture of experts.

From the training sample that we assumed to have in the previous section, we can form an estimate of $\hat{\theta}$ of θ. And this can be plugged into the posterior probability $\hat{f}(c|x;\hat{\theta})$ and this is usually termed as the plug-in classifier and is one of the most widely used classification methods. What is left to be decided is which estimator should be inserted in. The maximum likelihood (ML) estimator has been the most popular choice in statistics. The widespread use of the ML estimator in conjunction with the plug in approach arises from the excellent reputation that the ML estimator enjoys and because it has been directly or implicitly recommended by many of the pioneers in statistical classification theory.

The predictive approach as mentioned, is Bayesian in flavour and inspiration even though "vague prior" versions can be used and motivated outside the Bayesian paradigm. Assume now that we have a prior distribution for $f(\theta)$. Using Bayes theorem, we can generate a posterior density for θ, conditional on the data observed, $f(\theta|x)$. The main difference between the plug-in and predictive classifier is that the former acts as if the estimated θ is the true θ whereas the predictive approach averages over the uncertainty in θ. Hence

$$\hat{f}(c|x) = \int f(c|x;\theta) f(\theta|x) d\theta. \tag{1.3.1}$$

1.4 Mixture of Experts

The mixture of experts model was proposed by Jacobs and Jordan [1991] as a generalisation of mixture models, allowing for locally adaptive mixing weights. Mixture of experts comprises of a set of models, the experts, which performs the actual classification and a gating network which allocates the observations to different experts and averages (weighted) the predictions resulting from the experts.

Unlike so many other statistical methods that were invented/reinvented in the early 1990s, mixture of experts are not widely used. There seems to be several reasons. Firstly, due to non-convex nature of the likelihood, finding maximum likelihood estimates is considerably harder than for other statisti-

cal methods. And partly due to this difficulty, until recently, there has been no public general purpose implementation of mixture models. Furthermore, as they are generalisation of mixture models, their asymptotic properties are rather problematic. If the parameters of two experts are identical, the corresponding gating coefficients become unidentifiable and the parameter space collapses.

The outstanding feature of mixture of experts models is that they permit a meaningful interpretation of the model parameters while still being able to model complex relationships. Most other complex models, like neural networks and support vector-machines, are black boxes that they do not allow for a meaningful interpretation of their results. On the flip side, many simple statistical models which can be interpreted easily are usually too restrictive.

It is the structure of the mixture of experts that gives it its ease of interpretability. Similar to classification trees (CART) [Breiman et al., 1984], they partition the feature space and fit a simple model in each partition. Unlike CART, these splits are "soft". The partitioning of the covariate space is described by the parameters of the gating networks, which show where the boundaries are located and how "soft" they are and hence can be seen as defining different "regimes". The parameters of the experts can be interpreted as describing what these different regimes are like.

1.5 Outline of Thesis

Chapter Two discusses the issues in classifying partially labelled data using mixtures of normal and logistic regression and the data sets to be used in this thesis

Chapter Three defines hierarchical mixture of experts (HME) and compares it to related methods like classification trees, mixture models and neural networks.

Chapter Four discusses the frequentist fitting of the HME , different regularisation techniques and the results of applying our algorithm to the different data sets,comparing our results to the most widely used machine learning algorithms.

Chapter Five discusses the Bayesian fitting of HME and the MCMC techniques used generate the posterior density of the unknown parameters required.

Chapter Six discusses the advantages and disadvantages using logistic regression compared to normal distribution and an approach that combines the strengths of both.

Chapter Seven Conclusion and discussion for future research

Chapter 2

Classifying Partially labeled data

2.1 Data sets

There are six data sets to be used in this thesis for testing our algorithm :

2.1.0.1 *Leptograpsus* Crabs Data

Campbell and Mahon [1974] studied rock crabs of the genus *Leptograpsus*. One species *L. variegatus*, had been split into two new species, previously grouped by colour form, orange and blue. Preserved species lose their colour, so it was hoped that morphological differences would enable museum material to be classified.

Data are available on 50 specimens of each sex of each species, collected

on sight at Fremantle, Western Australia. For each species, we have measurements of the width of the frontal lip FL, the rear width RW, and length along the midline CL and the maximum width CW of the caraspace, and the body depth BD in mm.

To fit this data into our framework of unknown subclasses within a known class, two possible reasonable classification tasks can be considered: One is to classify the crabs into their colour forms assuming sex is unknown or classify them by sex, assuming that we do not know the colour form.

We have arbitrarily chosen to perform the latter classification task i.e. we have chosen to unlabel colour and treat it as the unknown subclass within the known class sex. Body depth is measured differently for females and it would seem prudent to remove it from the analysis.

As the data are physical measurements, we have chosen to work with log scale. A principal component analysis shows that the first principal component to be a "size effect" and since caraspace seems a reasonable proxy for size and hence all the measurements were divided by the carapace [Venables and Ripley, 2002]. It is also necessary to account for the sex differences, which we can do by analyzing each sex separately, or by subtracting the mean and the width of each sex which is what we did.

2.1.0.2 Music Data

This data set is taken from Weihs et al. [2006]. The task is to register classification i.e. correct labeling into high and low pitch of singers and

and instruments by pitch-independent features. The predictor variables are formed by using characteristics of the fundamental and the first 12 harmonics. The fundamentals [F0] of a sound is exactly its pitch frequency and the harmonics [F1,F2 ...] are integer multiples of the fundamental frequency. The pitch-independent variables are the mass of the harmonics F0-F12 of the fundamental frequency and the width (number of Fourier frequencies above some specific threshold in direct neighbourship to the harmonics in the normalised periodogram) without any information about its corresponding frequency. The data set comprises 432 observations (tone) played/sung by 9 different instruments/voices (the subclass).

2.1.0.3 Crystallisation Data

The crystallisation data set consists of 2746 observations. A total of 37 features were obtained from image analysis and the aim is to predict correctly whether an object is of crystalline nature or not, specified by 3 different classes. For more details on this dataset see Wilson [2006]. Each of the classes is known to be composed of seven subclasses.

2.1.0.4 Forensic Glass Data

The last data set to be used is described in Ripley [1996] where he describes the forensic testing of glass collected by B. German on 214 fragments of glass, Each glass possess measurements of its refractive index and chemical composition (weight percent of oxides of Na, Mg, Al, Si, K, Ca, Ba and

Fe). In the original classification, the fragments were classed as seven types, one of which was missing in this data set. The categories are: window float glass (70), window non-float glass (76), vehicle window class (17), containers (13), tableware (9) and vehicle headlamps (29). The composition sum up to approximately 100%.

In this case, the known main class is the type of glass and the unknown subclass is postulated to be the manufacturer of the glass. Note that the actual subclass is unknown and there could be more than one type of subclass within each known class or none at all. Professor Ripley understands from talking to an analytic chemist that for a given type of glass, different manufacturers produces glasses with markedly different chemical profiles. Hence our postulation that exists subclasses (which can be differentiated by the chemical profile of the glass) which represent the various different glass manufacturers appears to be a reasonable assumption.

Note that in this case, unlike the previous three data sets,the number of subclasses are unknown and would have to be estimated which is a more realistic scenario in real life because if the subclasses are known, it would be a fully labeled problem instead of a partially labelled problem

2.1.1 Waveform Data

This data set is taken from Breiman et al. [1984] and is used in Hastie and Tibshirani [1996] to illustrate mixture discriminant analysis. It is a three class problem and is considered a difficult pattern recognition problem. The

predictors are defined by

$$
\begin{aligned}
x_i &= uh_1(i) + (1-u)h_2(i) + \epsilon_i \quad \text{(class 1)}, \\
x_i &= uh_1(i) + (1-u)h_3(i) + \epsilon_i \quad \text{(class 2)}, \\
x_i &= uh_2(i) + (1-u)h_3(i) + \epsilon_i \quad \text{(class 3)}, \qquad (2.1.1)
\end{aligned}
$$

where $i = 1, 2, \ldots, 21$, u is uniform on $(0, 1)$, ϵ_i are standard normal variates and the h_i are shifted triangular waveforms: $h_1(i) = \max(6 - |i - 11|, 0)$, $h_2(i) = h_1(i - 4)$, $h_3(i) = h_1(i + 4)$,

2.1.2 Zip Code Data

This data set is described by LeCun et al. [1990] and is used in Hastie and Tibshirani [1996] to illustrate mixture discriminant analysis. It was chosen because it is a relatively simple machine vision task: The input consists of black and white pixels, the digits are usually quite well separated from the background, and there are only ten output categories. Yet the problem deals with image space to category space and the mapping from image space to category space has both considerable regularity and considerable complexity. And as Hastie and Tibshirani [1996] pointed out, it is an excellent candidate for subclass analysis as there are different ways of writing the same digit and this application has great practical applications.

The database consists of 9298 segmented numerals digitized from hand-

written zipcodes that appeared on the real U.S. Mail passing through Bufflao, N.Y. post office. The digits were written by many different people, using a great variety of sizes, writing styles and instruments and with widely varying levels of care. This was supplemented by a set of of 3349 printed digits coming from 35 different fonts.

We initially tried preliminary experiments on the larger 10 class problem but it was too computationally demanding and hence we decided to follow what Hastie and Tibshirani [1996] did and focus on the subproblem of distinguishing 3s, 5s and 8s.

2.2 Mixture Modelling

A mixture model can be defined as follows. Suppose there are n independent observations of features $x_1 \ldots, x_n$. Then under the mixture model , the density of x_i is given by:

$$f(x_i) = \sum_{w=1}^{W} \pi^w f^w(x_i), \tag{2.2.1}$$

Here π^w represents the prior probability of belonging to subclass w and $f^w(x_i)$ is the class conditional density of x_i given that it belong to subclass w. π^w are nonnegative quantities that sum to one i.e. that is

$$0 \leq \pi^w \leq 1 \tag{2.2.2}$$

and

$$\sum_{w=1}^{w=W} \pi^W = 1 \tag{2.2.3}$$

The quantities $\pi^1, \ldots, \pi^w$ are called mixing proportions or weights. As the functions $f^1 x_i, \ldots, f^w x_i$ are densities, it is obvious (2.2.1) defines a density. $f^w(x_i|x_i;\theta^w)$. are called component densities of the mixture. We will refer to (2.2.1) as a w-component finite mixture density. We will be only considering finite mixture models.

In this formulation of the mixture model, the number of components W is usually considered fixed. But in practice, in many applications, the value of W is unknown and has to be inferred from the available data, along with the mixing proportions and the parameters in the specified forms of the component densities

To help us interpret the mixture model. Consider this way of generating a random vector X_i with the $W-$component mixture density given by (2.2.1). Let Z_i be a categorical random variable taking on values $1, \ldots, W$ with probabilities $\pi^1, \ldots, \pi^W$ respectively and that the conditional density of X_i given $Z_i = w$ is $f^w(X_i)$ $(w = 1, \ldots, W)$. Hence the unconditional density of X_i (i.e. its marginal density) is given by $f(X_i)$. In this context, the variable X_i can be thought of as the component label of the feature vector X_i. Later

on it would be convenient to work with a W- dimensional component vector Z_i in place of a the single categorical variable Z_i, where the wth element of Z_i, $Z_i^w = (Z_i)^w$ is defined to be one or zero, according to whether the component of original of X_i in the mixture is equal to w or not. Thus Z_i is distributed according to multinomial distribution consisting of one draws on W categories with probabilities $\pi_1, \ldots, \pi_w$; that is

$$Pr\{(Z_i = z_i) = (\pi^1)^{z_i^1}(\pi^2)^{z_i^2}, \ldots, (\pi^W)^{z_i^W} \tag{2.2.4}$$

we write

$$Z_i \sim \text{Mult}_W(1, \pi) \tag{2.2.5}$$

In this interpretation of the a mixture situation, an obvious situation is where the w-component mixture (2.2.1) is directly applicable is where X_i is drawn from a population G which consists of W groups $G_1 \ldots G_W$ in proportions $\pi^1, \ldots, \pi^W$. If the density of X_i in group G_W is given by $f^w(x_i)$ for $w = 1, \ldots, W$ then the density of X_i has the w-component mixture form (2.2.1) in situation the W components of the mixture can be physically identified with the W external existing groups $G_1, \ldots, G_W$ as in our problem.

2.3 Incomplete-Data Structure of Mixture Problem

The idea of having a label vector Z^w associated with each feature vector X_i is a useful one, even though in a physical sense it might not be always appropriate to perceive the mixture in this sense. The usefulness of this conceptualisation of the mixture in this sense is that it allows the maximum likelihood estimate (MLE) of the mixture distribution to be computed via a straightforward application of the EM algorithm (which will be shown the following sections). It is also useful in implementing the MCMC methods in the fitting of mixture models in a Bayesian framework.

In this framework, the emphasis is on the estimation of mixture distributions on the basis of data, $x_1, \ldots, x_n$ usually available in the form of a observed random sample taken from the mixture density (2.2.1). That is $x_1, \ldots, x_n$ are the realised values of n independent and identically distributed (i.i.d) random vectors $X_1, \ldots, X_n$ with common density $f(x_i)$. We write

$$X_1, \ldots, X_n \overset{iid}{\sim} F, \tag{2.3.1}$$

where $F(x_i)$ denotes the distribution function corresponding to the mixture density $f(x_i)$.

Under the EM framework, the feature data $x_1, \ldots x_n$ are viewed as being

incomplete since the associated component-indicator vectors $z_1, \ldots, z_n$ are not available. The complete-data vector is hence declared to be

$$y = (x^T, z^T)^T. \tag{2.3.2}$$

where

$$y = (y_1^T, \ldots y_n^T)^T \tag{2.3.3}$$

is the observed-data or incomplete-data vector and where

$$z = (z_1^T, \ldots, z_n^T)^T \tag{2.3.4}$$

is the unobservable vector of component-indicator variables. We assume here that all the observations x_i have been completely recorded.

The components-label vectors $z_1, \ldots, z_n$ are taken to be realised values of random vectors $Z_1, \ldots, Z_n$ where, for independent feature data, it is appropriate to assume they are distributed conditionally as

$$Z_1 \cdots Z_n \overset{iid}{\sim} \text{Mult}_W(1, \pi),$$

The wth mixing proportion π^w can be interpreted as the prior probability that the observation belongs to the wth component of the mixture $w = 1, \ldots, W$ while the posterior probability that the observation belongs to the w component with x having been observed on it, is given by

$$\begin{aligned}\tau^w(y_i) &= pr\{Z_i^w = 1|x_i\} \\ &= \pi^w f^w(x_i)/f(x_i)(w = 1, \ldots W; i = 1, \ldots, n) \qquad (2.3.5)\end{aligned}$$

Later we will consider the formulation of an optimal rule of allocation in terms of these posterior of component membership

It can be observed that in this incomplete-data context, the mixture model arises because the component-label vectors are 'missing' from the complete-data vector and we have to estimate the mixture distribution on data available from the marginal distribution of X_i only rather than from the joint distribution of feature vector X_i and its component label Z_i. It will be seen later that the EM algorithm exploits this reduced simplicity of working with the joint distribution of X_i and Z_i to compute the MLES on the basis of the complete-data vector Y and then overcomes the fact that the label vectors z_i are unknown by iteratively working with the conditional expectation of the complete-data log likelihood given the observed data x, which is effected using the current fit of unknown parameters.

If the complete-data vector Y were available, estimation of mixture distri-

bution is straight forward than on the observed data y, since each component $f^w(y)$ could be estimated directly from the data known to have arisen from it, that is from those feature data x_i with $z_i^w = (z_i)^w = 1$. This would be trivial if say the components densities were hypothesised to be multivariate normal. The only other parameters then to be estimated would be the mixing proportions which in the case of a mixture sampling design for the classified data, can be estimated by proportion of these data from each component, namely

$$\hat{\pi}^w = \sum_{i=1}^{n} z_i^w / n \quad (w = 1, \ldots, W)$$

2.4 EM Algorithm

The EM Algorithm, introduced by Dempster et al. [1977], is an iterative technique for maximizing the likelihood in cases when there is no straightforward way of directly maximizing the likelihood. EM is an iterative algorithm and each iteration is composed of two steps:an Expectation (E) step and a Maximization (M) step. It starts with the observation that the optimization of the likelihood function will be greatly simplified if the values of a set of additional variables, called "missing" or "hidden" variables are known. The observable data set $\mathcal{X}$ is termed as the "incomplete data" set and we postulate a "complete data set" $\mathcal{Y}$ that includes the missing variables $\mathcal{Z}$. A probability model

that links the missing variables to the actual data is specified: $f(y, z|x, ; \theta)$. The logarithm of the density f defines the "complete-data loglikelihood" $l_c(\theta; \mathcal{Y})$. The original likelihood is $l_c(\theta; \mathcal{X})$ termed as the "incomplete-data likelihood". The EM algorithm starts by assuming some trial values of θ, the unknown parameters values that we wish to estimate. Next comes the E step where the expected value of the complete-data likelihood, given the observed data and the current model is computed :

$$Q(\theta, \theta^d) = E[l_c(\theta; Y|\mathcal{X})]. \tag{2.4.1}$$

where θ^d is the value of the parameter at the d iteration and the expectation is taken with respect to θ. This results in a deterministic function Q. Next comes the M step where this function is maximized with respect to θ to find the new parameter estimate θ^{pd+1} as:

$$\theta^{d+1} = \underset{\theta}{argmax}\ Q(\theta, \theta^d), \tag{2.4.2}$$

The E step is then repeated with the new parameter estimate θ^{d+1} to obtain an improved estimate of the complete data likelihood and the whole process repeats itself.

Each step of the EM yields a parameter value that increases the value of Q, the expectation of the complete likelihood. [Dempster et al., 1977] proved

that an increase in Q implies an increase in the incomplete likelihood:

$$l(\theta^{d+1}; X) \geq l(\theta^{d}; X). \tag{2.4.3}$$

Equality is only attained at the stationary pts of l [Wu, 1983]. While the EM algorithm generally does not guarantee a convergence to a global maximum, the likelihood l will never decrease with each step of the EM algorithm and in practice this leads to a convergence to a local maximum. The EM algorithm, however does have a reputation for very slow convergence [Redner and Walker, 1984].

2.5 Utilising the EM algorithm to Fit Mixtures

As previously discussed, in the formulation of the mixture problem in the EM framework, the observed-data vector

$$x = (x_1^T, \ldots, x_n^T) \tag{2.5.1}$$

is viewed as being incomplete, as the associated component labels vectors $z_1 \cdots z_n$ are not available. In this framework, where each x_i is conceptualised as coming from one of the components of the mixture model being fitted z_i^w is a w-dimensional vector with z_i^w=1 or 0 according to whether x_i did or did

not arise from the wth component of the mixture . The complete-data vector is therefore declared to be

$$y = (x^T, z^T)^T, \tag{2.5.2}$$

$$z = (z_1^T, \ldots, z_n^T)^T, \tag{2.5.3}$$

The component-labels of vectors $z_i, \ldots, z_n$ are taken to be realised values of the random vectors $Z_1, \ldots, Z_n$ where for independent feature data, we can assume that they are distributed unconditionally according to multinomial distribution (2.3.5). With this assumption, the distribution of the complete data vector Y implies the appropriate distribution of the incomplete-data vector X. The complete-data log likelihood for θ, $\log L_c(\theta)$ is given by

$$\log L_c(\theta) = \sum_{w=1}^{W} \sum_{i=1}^{n} z_i^w \{\log \pi^w + \log f^w(x_i, \theta^w)\} \tag{2.5.4}$$

The EM algorithm is applied to this problem by treating z_i^w as missing data. The addition of the unobservable data to the problem is handled by the E step which takes the conditional expectation of the complete-data log likelihood $\log L_c(\theta)$, given the observed data y, using the current fit for θ. Let $\theta^{(0)}$ be value specified initially for θ. Then on the first iteration

for EM algorithm, the E-step requires the computation of the conditional expectation of $\log L_c(\theta)$ given x, using $\theta^{(0)}$ for θ which can be written as

$$Q(\theta;\theta^{(0)}) = E_{\theta^{(0)}}(\log L_c(\theta)|x) \tag{2.5.5}$$

The expectation operator E has the subscript $\theta^{(0)}$ to explicitly convey that this expectation is effected using $\theta^{(0)}$ for θ. It follows on the $(d+1)$th iteration, the E step requires the calculation of $Q(\theta;\theta^{(d)})$, where $\theta^{(d)}$ is the value of θ after the dth EM iteration. As the complete-data log likelihood, $\log_c(\theta)$ is linear in the unobservable data z_i^w, the E step (on the $(d+1)$th iteration) simply requires the calculation of the current expectation of Z_i^w given the observed data x where Z_i^w is the random variable corresponding to z_i^w. Now

$$\begin{aligned} E_{(\theta^d)} &= \mathrm{pr}_{\theta^d}\{Z_i^w = 1|x\} \\ &= \pi_i^{w(d)} f^w(y_i;\theta^{w(d)})/f(y_i;\theta^{(d)}) \\ &= \pi^{w(d)} f^w(x_i;\theta^{w(d)}/\sum_v = 1^W \pi^{v(d)} f^v(y_i;\theta^{v(d)}) \\ &= \tau^w(y_i;\theta^{(d)}) \end{aligned} \tag{2.5.6}$$

for $w = 1,\ldots W;\ i = 1\ldots,n$. The quantity $\tau^w(x_i;\theta^{(d)})$ is the posterior probability that the wth member of the sample with observed value x_i belongs to the wth component of the mixture. Using (2.5.6) we obtain, on taking

the conditional expectation given x that

$$Q(\theta;\theta^{(k)}) = \sum_{w=1}^{W}\sum_{i=1}^{n}\tau^{w}(x_i;\theta^{(k)})\{\log\pi^{w} + \log f^{w}(f_i;\theta^{w})\} \tag{2.5.7}$$

The M-step on the $(d+1)$ iteration requires the optimisation of $Q(\theta,\theta^{(d)})$ with respect to θ over the parameter space Ω to compute the updated estimated $\theta^{(d+1)}$ for the finite mixture models, the updated estimates $\pi^{w(d+1)}$ of the mixing proportions π^{w} are independently calculated of the updated estimate $\xi^{(d+1)}$ of the parameters vector ξ containing the unknown vectors in the component densities.

If the z_i^w are observable, the complete-data MLE of π^w would be given simply by

$$\hat{\pi}^{w(d+1)} = \sum_{i=1}^{n} z_i^w/n \quad (w = 1,\cdots,W) \tag{2.5.8}$$

As the E step simply involves replacing each z_i^w with its current conditional expectation in complete-data log likelihood, the updated estimate of π is given by replacing each z_i^w in (2.5.8) by $\tau_i(y_i;\theta^{(d)})$ to give

$$\pi^{w(d+1)} = \sum_{i=1}^{n} \tau_i(y_i;\theta^(d))/n(w = 1,\ldots,W) \tag{2.5.9}$$

In forming the estimate π^w on the $(d+1)$th iteration, there exists a contribution from each observation x_i equal to its (currently assessed) posterior probability of membership of the wth component of the mixture model. It can be seen from that $\xi^{(d+1)}$ is obtained as an appropriate root of

$$\sum_{w=1}^{W}\sum_{i=1}^{n}\tau_i(y_i;\theta^{(d)})\partial\log f^w(x_i;\theta)/\partial\xi = 0 \tag{2.5.10}$$

A nice feature of the EM algorithm is that the solution of often exists in closed form as will be demonstrated for the normal mixture in the next section

2.6 Fitting Normal Distributions

Now we focus the on the case of a mixture of normal components,

$$f(y_i;\Psi) = \sum_{w=1}^{W}\pi^w\phi(x_i;\mu^w,\Sigma^w) \tag{2.6.1}$$

where $\phi()$ is the multivariate normal distribution. Here $\Psi = (\pi^1,\ldots,\pi^{w-1},\xi^T)^T$ where ξ contains the elements of the components means μ^w and the distinct elements of the component-covariance matrices Σ^w $(w = 1,\ldots,W)$. We shall first consider the unrestricted (heteroscedastic) case where the component variances are unequal i.e. there are no restrictions placed on them.

On the $(d+1)$th iteration of the E step, where the zero-component-label

variables z_i^w are replaced by their current conditional expectations given by the posterior probabilities of component membership of the observed data x_i $\tau^w(y_i;^{(d)})$, where

$$\tau^w(x_i;\Psi) = \pi^w \phi(x_i;\mu^w,\Sigma^w)/\sum_{v=1}^{W}\phi(x_i;\mu^v\Sigma^v)$$

for $w = 1,\ldots,W$; $i = 1,\ldots,n$

The M-step for normal components exists in closed form. The updates for the component means μ^w and component-covariances matrices Σ^W are given simply by

$$\mu^{w(d+1)} = \sum_{i=1}^{n}\tau_i^{w(d)}x_i/\tau_i^{w(d)} \tag{2.6.2}$$

and

$$\Sigma^{w(d+1)} = \sum_{i=1}^{n}\tau_i^{w(d)}(x_i-\mu^{w(d+1)})(y_i-\mu^{w(d+1)})^T/\sum_{i=1}^{n}\tau_i^{w(d)} \tag{2.6.3}$$

for $w = 1,\ldots W$, where

$$\tau_i^{w(d)} = \tau^w(x_i;\Psi^{(d)}) \quad w = 1,\ldots,w = W;\ i = 1,\ldots,n. \tag{2.6.4}$$

The updated estimate of the wth mixing proportion π^w is given by (2.5.9)

It is computationally advantageous to express the update (2.6.4) of Σ^w directly in terms of the current conditional expectations of the sufficient statistics T_1^w, T_2^w and T_3^w for Ψ in the complete-data framework given by

$$T_1^{w(d)} = \sum_{i=1}^{n} \tau_i^{w(d)},$$
$$T_2^{w(d)} = \sum_{i=1}^{n} \tau_i^{w(d)} x_i,$$

and

$$T_3^{w(d)} = \sum_{i=1}^{n} \tau_i^{w(d)} y_i y_i^T$$

and hence we have

$$\Sigma^{w(d+1)} = \{T_3^{w(d)} - (T_1^{w(d)})^{-1} T_2^{w(d)} (T_2^{w(d)})^T\} \quad w = 1, \ldots, W \qquad (2.6.5)$$

using (2.6.5) instead of (2.6.4) gives a reduction in CPU time of around 50%

Often in practice, the component covariance matrices σ^W are restricted to being equal,

$$\Sigma^W = \Sigma \qquad (w = 1, \ldots, W) \tag{2.6.6}$$

where Σ is unspecified. Under the case of homoscedastic normal components, the updated estimate Σ is given by

$$\Sigma^{(d+1)} = \sum_{w=1}^{W} T_1^{w(d)} \Sigma^{w(d+1)}/n, \tag{2.6.7}$$

where $\Sigma^{w(d+1)}$ is given by (2.6.5) and the updates of π^w and μ^w are as above in the heterodscedastic case

2.7 Classifying partially labeled data using mixture of normals

Now we look at classifying partially labeled data using mixture of normals and the notable work is done by Hastie and Tibshirani [1996]. In a classification problem, the outcome of interest y falls into C unordered classes which we denote by the set $B = \{1, 2, 3, \ldots, C\}$. Suppose we possess training data (x_i, y_i), $i = 1, 2, \ldots, n$. Each class c is divided into W subclasses. The model assumes that each subclass has a multivariate normal distribution with its own mean vector μ^{cw} and common variance matrix Σ. As highlighted earlier,

it is not necessary for the covariance matrix to be equal but they adopted this model as it keeps the total of number of parameters under control and allows them to permit the other generalisations that they have in mind later on.

Let Π^c be the prior probability for class c and within class c and let π^{cw} be the mixing probability for the w subclass,

$$\sum_{w=1}^{W} \pi^{cw} = 1 \tag{2.7.1}$$

Although often the Π^c are known or easily estimated from the training data, the π^{cw} are unknown parameters. Let

$$D(x, \mu) = (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{2.7.2}$$

be the Mahalanobis distance between x and μ.

The mixture density of class c is

$$f^c(x) = f(x|y = c) \tag{2.7.3}$$

$$= |2\pi\Sigma|^{-1/2} \sum_{w=1}^{W} \pi^{cw} \exp\{-D(x, \mu^{cw}/2)\}, \tag{2.7.4}$$

and the conditional log-likelihood for the data is

$$l^{mix}(\mu^{cw}, \Sigma, \pi^{cw}) = \sum_{i=1}^{n} \log f_{y_i}(x_i). \tag{2.7.5}$$

The EM algorithm provides a straightforward way for maximising $l^{mix}(\theta)$. The EM steps are

$$\begin{aligned} f(b^{cw}|x, c) &= Prob(x \in w\text{th subclass of class } c|x, c) \\ &= \frac{\pi^{cw} \exp\{-D(x, \mu^{cw})/2\}}{\sum_{w=1}^{W} \exp\{-D(x, \mu^{cw})/2\}} \end{aligned} \tag{2.7.6}$$

$$\pi^{cw} \propto \sum_{y_i=c} f(b^{cw}|x_i, c), \quad \sum_{w=1}^{W} \pi^{cw} = 1 \tag{2.7.7}$$

$$\mu^{cw} = \frac{\sum_{y_i=c} x_i f(b^{cw}|x_i, c)}{\sum_{y_i=j} f(b^{cw}|x_i, c)} \tag{2.7.8}$$

$$\Sigma = \frac{1}{N} \sum_{c=1}^{C} \sum_{y_i=c} \sum_{w=1}^{W} f(b_{cw}|x_i, c)(x_i - \mu^{cw})(x_i - \mu^{cw})^T \tag{2.7.9}$$

The notation $\sum_{y_i=c}$ means summing over all observations belonging to the cth class.

In the above, the expression (2.7.6) is the estimation step, whereas expressions (2.7.7) - (2.7.9) are the maximisation steps. This is a straightforward generalisation of the EM algorithm presented earlier for one mixture. Expressions (2.7.8)- (2.7.9) have the same expression as the maximum likelihood estimate for the complete normal discriminant problem i.e. the situation where we observe the subclass membership. The only difference is that the subclass indicator is replaced by $f(b^{cw}|x_i, c)$, the estimated probability the observation i falls into sublcass b^{cw}. Equations (2.7.6) - (2.7.9) are iterated till a suitable terminating convergence criteria.

The posterior class probabilities are

$$f(f_i = c|X = x) \sim \Pi^c f(x|c) \sim \Pi^c \sum_{w=1}^{W} \pi^{cw} \exp\{-D(x, \mu^{cw})/2\} \qquad (2.7.10)$$

normalised so that

$$\sum_{c=1}^{C} f(y_i = c|X = x) = 1 \qquad (2.7.11)$$

The classification rules chooses c to maximize $f(c|x)$.

2.8 Classifying partially labeled data using mixture of logistic regressions

Now we look at statistical method problem that is the focus of this thesis, classifying partially labeled data using mixtures of logistic regressions. Assuming that we that we know the subclass w, the probability of observation y_i belonging to class c is:

$$f(y_i = c|x) = \frac{\exp(\theta^{cw} x_i)}{\sum_{u=1}^{C} \sum_{v=1}^{V} \exp(\theta^{uv} x_i)}. \tag{2.8.1}$$

Since we do not actually know the subclass w, we will have to sum over all possible subclasses w within the class c. Formally our classifier is:

$$f(y_i = c|x_{n+1}; \theta) = \sum_{w=1}^{W} \frac{\exp(\theta^{cw} x_i)}{\sum_{u=1}^{C} \sum_{v=1}^{W} \exp(\theta^{uv} x_i)}. \tag{2.8.2}$$

Using the forensic glass as an example, c is the type of glass (window float,vehicle, etc), the reported and known class, w represents the manufacturer, the subclass within the known main class c (type of glass).

2.9 Determining the number of components of a mixture

Determining the number of components/clusters in a mixture, often labeled as k as in the k-means algorithm is a frequent problem in data clustering and is a distinct issue from the process of optimising the mixture.

For certain classes of algorithms, in particular K-means, K-medoids and the EM algorithm, there is a parameter commonly referred to as k that specifies the number of clusters to detect. Other algorithms such as DBSCAN and OPTICS algorithm does not require the specification of this parameter. Hierarchical clustering avoids the problem altogether.

The optimal choice of k is often ambiguous with and depends on the shape of the distribution of the data set and desired resolution specified by the user. Increasing k without penalising model complexity will always reduce the error rate for data the model in fitted, eventually producing the saturated model in which each data point belongs to its own cluster.

2.9.1 The Elbow Method

This method looks examines the percentage of variance explained as a function of number of clusters. The number of clusters is increased until adding another cluster does not significantly improve the modeling of the data. If one plots the percentage of variance against the number of clusters, the first cluster will yield the greatest reduction in variance (give the most informa-

tion). Successive clusters will yield smaller decreases in variance but at some point, the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point hence the name "elbow criterion". It is not always possible to unambiguously identify this "elbow" Ketchen and Shook [1996]. Percentage of variance explained is the ratio of between-group variance to the total variance and can be tested using F test. A slight variation of this algorithm plots the curvature of the within group variance. Goutte [1999]. This algorithm can be traced to speculation by Thorndike [1953].

2.9.2 The Silhouette Method

The average silhouette of the data can also be another useful criteria for assessing the optimal number of clusters. The silhouette of a datum is a measure of how closely it is matched to data within its cluster and how loosely it is matched to data of neighbouring cluster i.e. the cluster whose average distance from the datum is the lowest Rousseeuw [1987]. A silhouette value close to 1 implies that the datum is in an appropriate cluster, while a silhouette close to -1 implies the datum is in the wrong cluster. Optimisation algorithms such as genetic algorithms can be utilised in determining the number of clusters that gives rise to the largest silhouette Lleti et al. [2004].It is also possible that to rescale the data in such a way that the silhouette is more likely to be maximised at correct number of clusters de Amorim and Hennig [2015]

2.9.3 Cross-Validation

The data is divided into k folds. Each fold is set aside as a test set and a clustering model is computed on the other $k-1$ training sets and the value of the goal function (for example, the sum of squared distances to the centroids for k-means) calculated for each test set. The k values are calculated averaged for alternative number of clusters and the cluster number selected is the one that minimises the test set error.

2.9.4 Bayesian Approach

Lenk and DeSarbo [2000] suggested computing the posterior probabilities of the number of mixture components using MCMC. The number of mixture components can be selected by choosing the model with the largest posterior probability. If the number of components are a priori equally likely, then model with the largest Bayes factor is chosen. Both procedures require computing the marginal density of the data given the number of mixture components.

2.9.5 Information Criterion Approach

If a likelihood function for the clustering model can be formed, then information criteria such as Akaike information criteria (AIC), Bayesian information criteria (BIC) can be used to determine the number of clusters.

In many studies related to model selection, it is discovered that AIC

may select too large and model and BIC may select too small a model. This phenomenon appear to hold true as well in selecting K in the mixture analysis. Jeffrey D. Banfield [1993] proposed using approxmiate weight of evidence as an approximate Bayesian model selection. Some empirical studies seem to favour the use of BIC [Fraley. and Raftery, 1998]. This is the approach we will pursue here. The Mclust package in R [Fraley et al., 2012], implementing model based clustering has a function for choosing the number of clusters by BIC and this is the method we will adopt here.

2.9.6 An Information Theroetic Approach

Rate distortion theory has been applied to determining the number of clusters using the "jump" method which determines the number of clusters that maximizes efficiency while minimising error by information theoretic standards Sugar and James [2003]. The strategy is to generate a distortion curve for the input data by running a standard clustering algorithm such as k-means for all values between 1 and n and computing the distortion (described below) of the resulting clustering. The distortion curve is then transformed by a negative power chosen based on dimensionality of the data. Jumps in the resulting values signify reasonable choices for k with the largest jump representing the best choice.

The distortion of a clustering algorithm is formally defined as follows: Let the data set be modeled as a p -dimensional random variable X consisting of a mixture distribution of W components with a common covariance Σ.

Suppose now we set $c_1, \ldots, fIc_k$ be a set of K cluster centers, with C_X the closest center to a given sample of X, then the minimum average distortion per dimension when fitting K centers to the data is

$$d_K = \frac{1}{p} \overset{min}{c_1, \ldots, c_k} E[(X - c_X)^T \Sigma^{-1} (X - c_X)] \tag{2.9.1}$$

This coincides with the average Mahalanobis distance between X and the set of cluster centers C. As the minimisation over all possible sets of clusters is prohibitively complex, the distortion is in practice computed by generating a set of cluster centers using a standard clustering algorithm and computing the distortion using the expression above. The pseudo-code for the jump method with an input set of p dimensional data points X is:

JumpMethodX

Let $Y = (p/2)$

Init a list D, of size $n + 1$

Let $D[0] = 0$

For $k = 1, \ldots, n$;

Cluster X with k clusters

Let $d =$ Distortion of the resulting clustering

$D[k] = d^{(-Y)}$

Define $J(i) = D[i] - D[i - 1]$

Return the k between 1 and n that maximise $J(k)$

The choice of the transform power $Y = (p/2)$ is motivated from asymptotic reasoning using results from distortion theory. Let the data X have a single arbitrary p dimensional Gaussian distribution and let fixed $K = \lfloor \alpha^p \rfloor$ for α greater than zero. It can be demonstrated asymptotically that the distortion of a clustering to the power $(-p/2)$ is proportional to α^p which by definition is approximately the number of clusters K. Hence for a single Gaussian distribution, increasing K beyond the true number of clusters, which should be one, causes a linear growth in distortion. This behaviour is crucial in the general case of a mixture of multiple components.

Let X be a mixture of W p-dimensional Gaussian distributions with common covariance. Then for any fixed K less than W, the distortion of a clustering as p goes to infinity is infinite. This implies that a clustering of less than the correct number of clusters is unable to describe asymptotically high-dimensional data, resulting the distorting increasing without limit. Since K is an increasing function of p i.e. $K = \lceil \alpha^p \rceil$, the same result as above is attained, with the value of the distortion in the limit as p goes infinity equal to α^{-2}. Correspondingly, there is the same proportional relationship between the transformed distortion and the number of clusters.

As a result, it can be observed that for sufficiently high values of p, the transformed distortion $d_K^{-p/2}$ is approximately zero for $K < W$, then jumps suddenly and begins increasing linearly for $K > W$. The jump algorithm for choosing K makes use of these behaviours to identify the most likely value for the true number of clusters.

Although the mathematical justification for the method is given in terms of asymptotic results, the algorithm has been empirically demonstrated to work well in a variety of data sets with reasonable dimensionality. In addition to the localised jump algorithm described above, there exists another algorithm for choosing K using the same transformed distorted values known as the "broken line" method. This method identifies the jump point in the graph of the transformed distortion by performing a simple least squares error line fit of two line segment, which in theory will fall along the x-axis for $K < G$, and along the linearly increasing phase of the the transformed distortion plot of $K \geq W$. This broken line method is more robust than the jump method in that its decision is global rather than local, but it has the disadvantage of relying on the assumption of Gaussian mixture components, whereas the jump method is fully non-parametric and has been shown to be viable for general mixture distributions.

Chapter 3

Mixture of Experts

3.1 Introduction

The model that we have chosen to use in this thesis is the mixture of experts developed by Jacobs and Jordan [1991]. The idea was inspired by Hampshire et al. [1992] and the motivation behind it was to reduce interference effects in neural networks: the performance in one part of the data space is very likely to deteriorate when trying to optimise the parameters for another part for the data space. The gist of Jordan's idea was to train experts to focus on the different parts of the data space and a gating network is trained to allocate different observations to different experts.

3.2 Mixture of Experts

Adaptive mixture of experts (ME), developed by Jacobs and Jordan [1991] can be defined in the following manner. Suppose there are n independent observations of response $y_1 \dots y_n$ with associated features $x_1, \dots, x_n$. Then under the ME model, the probability of y_i given feature x_i is given by:

$$f(y_i|x_i) = \sum_{j=1}^{J} g_i^j(x_i;\gamma^j) f^j(y_i|x_i;\theta^j), \tag{3.2.1}$$

where (γ^j, θ^j) denotes the set of unknown parameters that we wish to estimate.

In the literature of machine learning, ME is described in terms of two underlying networks, gating networks and expert networks. Within each region of the feature space, the expert networks (there are J of them here) approximate the distribution of the response variable by mapping the the feature vector x_i to an output, the density $f^j(y_i|\theta^j)$. Different experts are assumed to be suitable for different regions of the feature space and here is where the gating networks come in. The gating networks utilise the feature vector to identify the expert or the combination of experts whose output best approximates the corresponding density $f(y_i)$ of the response y_i . The gating network produces a set of scalar coefficients (gating weights) $g_i^j(x_i;\gamma^j)$, which weigh the contributions from the various experts. From the perspective of statistical mixture modelling, the gating networks model the input-dependent mixing probabilities, which are the probabilities in a multinomial distribution

consisting of one draw on J categories and the expert networks model input-dependent component densities $f^j(y_i|\theta^j)$.

3.2.1 Hierarchical Mixture of Experts(HME)

The ME model only has one level of experts. Jordan and Jacobs [1992] demonstrated empirically that models with more than one level of experts often perform better than single-level models, given the same number of free parameters. When there are two or more levels of experts, we have the hierarchical mixture-of-experts (HME) model proposed Jordan and Jacobs [1992]. It takes the following form:

$$f(y_i|x_i) = \sum_{j=1}^{J} g_i^j(x_i;\gamma^j) \sum_{k=1}^{K_j} g_i^{k|j}(x_i;\gamma^{jk}) f^{jk}(y_i|x_i;\theta^{jk}), \tag{3.2.2}$$

where $(\gamma^k, \gamma^{jk}, \theta^{jk})$ is the vector of unknown parameters. $g_i^j(x_i;\gamma^j)$ is the probability of assigning an observation y_i with feature x_i to the jth component of the first level. $g_i^{k|j}(x_i;\gamma^{jk})$ is the conditional probability that observation y_i (with feature x_i) belongs to the kth component of the jth component of the first level, given that it belongs to the jth component of the first level.

The gating networks usually take the form of mutliclass logistic regressions:

$$g_i^j(x_i;\gamma^j) = \frac{\exp(\gamma^j x_i)}{\sum_{r=1}^{J} \exp(\gamma^r x_i)} \quad j = 1, \ldots, J$$

and

$$g_i^{k|j}(x_i;\gamma^{jk}) = \frac{\exp(\gamma^{jk}x_i)}{\sum_{s=1}^{K_j}\exp(\gamma^{js}x_i)} \quad k = 1,\ldots,K_j.$$

At each expert (terminal node), we have a model for the response variable of the form:

$$Y \sim f^{jk}(y_i|x_i,\theta^{jk})$$

and this will vary according to the problem and is usually a generalised linear model (GLM).

The experts in our case is the multiclass logistic regression introduced in chapter one: Equation (2.8.2). Hence the full likelihood of our problem is:

$$f(y_i = c|x_i) = \sum_{j=1}^{J}\frac{\exp(\gamma^{j}x_i)}{\sum_{r=1}^{J}\exp(\gamma^{r}x_i)}\sum_{k=1}^{K_j}\frac{\exp(\gamma^{jk}x_i)}{\sum_{s=1}^{K_j}\exp(\gamma^{js}x_i)}\sum_{w=1}^{W}\frac{\exp(\theta^{jkcw}x_i)}{\sum_{u=1}^{C}\sum_{v=1}^{W}\exp(\theta^{jkuv}x_i)}.$$

Shown in figure (3.1) is a graphical representation of a HME.

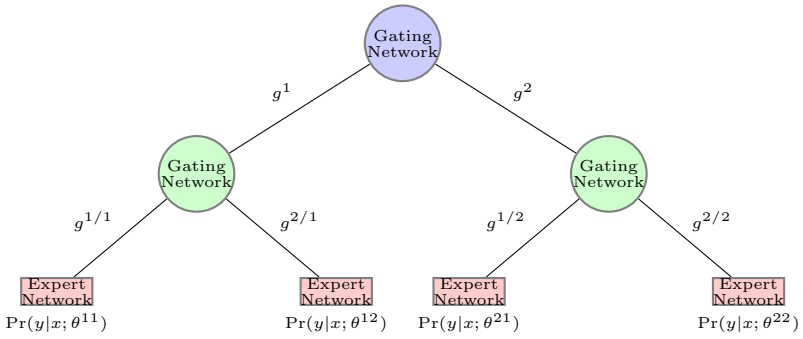


Figure 3.1: A two-level hierarchical mixture of experts

3.3 Comparison to Related Classification Methods

3.3.1 Classification trees

Classification trees are a very popular alternative to mixture of experts. Both of them are predicated on the idea of partitioning the problem into small subproblems which can be solved using a fairly simple model. In HME, a hierarchical logistic regression is used to partition the data space and in each partition a GLM is used. Regression trees [Breiman et al., 1984] takes this approach to the extreme. The response is constant in each of the partitions.

One main difference between classification trees and mixture of experts is the way the models are fitted. For HME, when the HME tree structure is already specified a priori, one attempts to find the maximum likelihood of the gating and expert parameters which is a non-trivial task due to the

complex nature of the likelihood. In general, classification trees do not even try to find the globally optimal tree. They merely utilise a greedy "one step look ahead" strategy. Starting with a single partition, further partitions are recursively split. As a split can only be carried out parallel to the coordinate axes, the optimal split is obtained by trying every possible split. Once a split has been performed, it cannot be changed anymore. This makes the algorithm extremely easy to implement. However, there is no guarantee of any global optimality. In the CART algorithm, the selection of the correct tree structure is an integral part of the algorithm i.e. the CART algorithm "grows" the tree itself. This step is crucial as splits which have already been performed cannot be undone. For the HME, the gating tree that is used needs to be specified a priori. Growing the the trees is not as important for HME models as it is for CART-like models as the estimation is performed using maximum likelihood i.e. a globally optimal solution is found. In the next chapter we would address how an optimal tree can be grown for HME. Coupled with the fact that decision trees use hard splits and a constant model in each partition gives decision trees its ease of interpretability. The partitioning of the feature space can be described by a single tree. This representation is highly favoured by medical scientists, probably because it is akin to the way a doctor thinks. The tree stratifies the population into strata of high or low probability (of outcomes) based on the symptoms of the patients. [Hastie et al., 2001].

The big disadvantage of classification trees is they tend to be of high

variance. Frequently, a minor change in the data can result in a vastly different series of splits, making interpretation difficult. It is the hierarchical nature of the process that results in this instability. An error in the top split is propagated down to all subsequent splits. One possible solution is to use a more stable split criterion but it does not remove the inherent instability: it is an inevitable price to be paid for a simple tree-based structure [Hastie et al., 2001]. Another possible solution which is employed by CART, MART, ID3, mixtures and mixture of experts models is to fit piecewise constant or piecewise linear functions. This approach minimizes variance at a cost of increased bias [Jordan and Jacob, 1994].

Mixture of experts also attempts to reduce the variance by using soft splits. In a hard split, an observation is assigned exclusively to a certain partition whereas in a soft split, the probability that an observation belongs to a certain partition is modelled [Bridle, 1989, Nowlan, 1991, Wahba et al., 1993]. This approach, by allowing data to lie simultaneously in multiple regions, permits parameters in one region to be influenced by data in neighbouring regions. Tree based methods adopt hard splits, which has severe effects on variance as information from neighbouring partition are not used at all. By permitting soft splits, the severe effects of chopping off distant data can be mitigated somewhat. Mixture of experts models also try to minimize bias that is incurred by utilizing piecewise linear functions, by permitting the splits to be formed along hyperplanes at arbitrary orientations in the input space. [Jordan and Jacob, 1994]

3.3.2 Neural Networks

Mixture of experts arose out of a desire to reduce interference in neural networks [Jacobs and Jordan, 1991]

The output (prediction) of a feed-forward neutral network with a single linear output node and a single hidden layer of K nodes is:

$$\sum_{k=1}^{K} v^k h_k(x_i w^k). \tag{3.3.1}$$

In contrast, the prediction of a mixture of experts model is:

$$f(y_i|x_i) = \sum_{j=1}^{J} g_i^j(x_i;\gamma^j) f^j(y_i|x_i;\theta^j). \tag{3.3.2}$$

One of the main differences between the two models is that the weights $g_i^j(x;\gamma^j)$ used in the HME model depends on the observation index i whereas the weights v^k used in a neural network are global. In an HME model, when the parameter θ^j is changed, only observations for which $g_i^j(x;\gamma^j)$ is not (close to) 0 will be affected whereas in a neural network changing w^k will affect all observations. Hence trying to optimise w^k for some part of the data space might result in deterioration of the performance of the neural network in other parts of the data space and this phenomenon is termed as interference. This problem does not exist for mixture of experts because of the localised weights as the regression parameters for non-neighbouring experts are decoupled.

The other main advantage of mixture of experts, over black box methods like neural networks, is that it allows for a meaningful interpretation of the model parameters while still retaining the ability to model complex relationships.

3.3.3 Mixture Models

Although mixture of experts arose out of a desire to reduce interference in neural networks, it is actually a generalisation of a mixture model. The main difference between mixture of experts and mixture models is that the mixing weights do not depend on the different covariates of the observations i.e. the different observations have the same weight of belonging to the same subclass. Hence HMEs can be utilized for curve fitting whereas mixture regression models cannot be as the overall mean would still be a linear function of the design matrix. While classical mixture regression models are used for modelling heterogeneity in the population studied, HME models are mostly used to model local changes in regression parameters. However since mixture models are merely a special case of HME, HME can be used in situations where mixture models are used. This can be really useful when heterogeneity is observed in some part of the population or when studying multi-valued prediction problems.

Chapter 4

Maximum Likelihood Estimation of Mixture of Expert Models

4.1 Introduction

Mixture of experts is a mixture model and hence the EM algorithm can be used to fit it. Shortly after Jacobs and Jordan [1991] came up with the mixture of experts model, Jordan and Jacob [1994] proposed using the EM algorithm to fit the HME.

4.2 Posterior Probabilities

For the learning algorithms to be utilised in subsequent sections, it will prove useful to develop the posterior probabilities associated with the nodes of the trees. The terms "posterior" and "prior" have meaning in this context during the training of the system. The probabilities $g^j(x)$ and $g^{k|j}(x)$ are referred to as prior probabilities, because they are computed based only on the input x, without knowledge of the corresponding output y. The posterior probability is defined once both the input and the target output are known. Using Bayes rule, the probabilities at the nodes of the trees are defined as follows:

$$h^j = \frac{g^j \sum_s g^{s|j} f^{js}(y)}{\sum_r g^r \sum_s g^{s|r} f^{rs}(y)} \tag{4.2.1}$$

and

$$h^{k|j} = \frac{g^{k|j} f^{jk}(y)}{\sum_s g^{s|j} f^{js}(y)}. \tag{4.2.2}$$

It will also useful to define the joint posterior probability h^{jk}, the product of h^j and and $h^{k|j}$

$$h^{jk} = \frac{g^j g^{k|j} f^{jk}(y)}{\sum_r g^r \sum_s g^{s|r} f^{rs}(y)}. \tag{4.2.3}$$

This quantity is the probability that the expert network (j, k) is considered to have generated the data, based on both the knowledge of the input and the

output. Again, it is to be emphasised that all these quantities are conditional on the input x.

For trees with more than two levels, the posterior probability associated with an expert network is simply the product of the conditional posterior probabilities along the path from the root of the tree to the expert.

4.3 EM Algorithm for fitting HME

For the application of the EM algorithm to our problem, appropiate "missing" data must be defined so as to simplify the likelihood function. There are two different type of mixtures here, one for the overall HME architecture and one for the experts and we will look at the HME architecture first. For the HME architecture, we introduce indicator variables z_i^j and $z_i^{k|j}$ where z_i^j is such that one and only one of z_i^j is equal to one and one and only one of $z_i^{k|j}$ is equal to one. These indicator variables have an interpretation as the labels that correspond to the decisions in the probability model. z_i^j is one if the observation y_i belongs to the jth component of the first level of the HME and zero otherwise. $z_i^{k|j}$ is one if y_i belongs to the kth component of the second level of the HME, given that it belongs to jth component of the first level of the HME and zero otherwise. The indicator variable z_i^{jk} is defined as the product of these two indicator variables. This indicator variable has an interpretation as the label that specifies the expert in the probability model.

With the indicator variables, the probability model for our problem can

be written in terms of the z_i^{jk}s as follows:

$$f(y_i, z_i^{jk}|x_i;\theta) = \prod_{j=1}^{J}\prod_{k=1}^{K_j}\left[g_i^j g_i^{k|j} f^{jk}(y_i)\right]^{z_i^{jk}}, \tag{4.3.1}$$

making use the of the fact that z^{jk} is an indicator variable. We then take logarithms of this probability model to yield the complete-data log-likelihood

$$\begin{aligned} l_c(\theta;\mathcal{Y}) &= \sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{K_j} z_i^{jk}\ln\{g_i^j g_i^{k|j} f^{jk}(y_i)\} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{K_j} z_i^{jk}\{\ln g_i^j + \ln g_i^{k|j} + \ln f^{jk}(y_i)\}. \end{aligned} \tag{4.3.2}$$

Next we calculate the expectation of the indicator variable z_i^{jk}:

$$\begin{aligned} E[z_i^{jk}|\mathcal{X}] &= f(z_i^{jk}=1|y_i,x_i,\theta^d) \\ &= \frac{f(y_i|z_i^{jk}=1,x_i;\theta^d)f(z_i^{jk}=1|x_i;\theta^d)}{f(y_i|x_i;\theta^d)} \\ &= \frac{f^{jk}(y_i)g_i^j g_i^{k|j}}{\sum_{r=1}^{J} g_i^r \sum_{s=1}^{K_j} g_i^{s|r} f_i^{rs}(y_i)} \\ &= h_i^{jk} \quad \text{(Equation (4.2.3))}. \end{aligned} \tag{4.3.3}$$

Next we look at the case of indicator variables for the experts. We introduce another indicator variable z_i^{jkcw} where jk denotes membership for the expert as above and c denotes membership to the labeled class and w denotes membership to the unlabelled class within the known class c. The thing to note

is that the class c is known and is given by the training sample and the and it is w which is unknown and is what we are imputing.

With the indicator variable, the probability model for the expert can be written in terms of the z_i^{jkcw}s as follows:

$$f^{jk}(y_i, z_i^{jkcw}|x_i;\theta) = \left[\prod_{w=1}^{W} \frac{\exp(\theta^{jkcw}x_i)}{\sum_{u=1}^{C}\sum_{v=1}^{W}\exp(\theta^{jkuv}x_i)}\right]^{z_i^{jkcw}}, \tag{4.3.4}$$

making use the of the fact that z^{jkcw} is an indicator variable. Taking logs we have:

$$\ln f^{jk}(y_i, z_i^{jkcw}|x_i;\theta) = \sum_{w=1}^{W} z_i^{jkcw} \ln\left[\frac{\exp(\theta^{jkcw}x_i)}{\sum_{u=1}^{C}\sum_{v=1}^{W}\exp(\theta^{jkuv}x_i)}\right] \tag{4.3.5}$$

Next we calculate the expectation of the indicator variable z_i^{jkcw}:

$$\begin{aligned} E[z_i^{jkcw}|\mathcal{X}] &= f(z_i^{jkcw} = 1|y_i, x_i;\theta^d) \\ &= \frac{\frac{\exp(\theta^{jkcw}x_i)}{\sum_{u=1}^{C}\sum_{v=1}^{V}\exp(\theta^{jkuv}x_i)}}{\sum_{s=1}^{W}\frac{\exp(\theta^{jkcs}x_i)}{\sum_{u=1}^{C}\sum_{v=1}^{W}\exp(\theta^{jkuv}x_i)}} \\ &= h_i^{jkcw}. \end{aligned} \tag{4.3.6}$$

Substituting in the expected values of the indicator variables for the HME architecture (we have delayed the substitution of the expected values of the

indicator variables for the experts till a later step):

$$Q(\theta, \theta^d) = \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{k=1}^{K_j} h_i^{jk} \{\ln g_i^j + \ln g_i^{k|j} + \ln f^{jk}(y_i)\}. \tag{4.3.7}$$

Next comes the M step where we maximize $Q(\theta)$ with respect to the expert and gating network parameters and because of the use of the indicator variables, the optimisation problem decomposes into three separate independent optimisation problems, one for each level of gating networks and for the experts (here we have substituted in the expected values of the indicator variables for the experts):

$$\begin{aligned}
(\gamma^j)^{d+1} &= \underset{\gamma^j}{argmax} \sum_{i=1}^{n} \sum_{j=1}^{J} h_i^j \ln g_i^r \\
(\gamma^{jk})^{d+1} &= \underset{\gamma^{jk}}{argmax} \sum_{i=1}^{n} \sum_{k=1}^{K_j} h^{jk} \ln g_i^{k|j} \\
(\theta^{jkcw})^{d+1} &= \underset{\theta^{jkcw}}{argmax} \sum_{i=1}^{n} h_i^{jk} \sum_{w=1}^{W} h_i^w \ln \left[\frac{\exp(\theta^{jkcw} x_i)}{\sum_{u=1}^{C} \sum_{v=1}^{W} \exp(\theta^{jkuv} x_i)} \right],
\end{aligned}$$

and the whole process reiterates using the updated parameter values until it satisfies some suitable terminating condition.

4.4 Advances in the EM algorithm

Other than slow convergence, the other main issue with the EM algorithm is that it only guarantees convergence to local maximum and not a global

one. We will address this issue later. We will first highlight some of the advances in the EM algorithm, especially with respect applying it to ME models. For ease of exposition we will assume a ME model instead of a HME. The equation for the ME model is

$$f(y_i|x_i) = \sum_{j=1}^{J} g_i^j(x_i;\gamma^j) f^j(y_i|x_i;\theta^j) \tag{4.4.1}$$

To apply the ME algorithm to the ME networks, the indicator variable z_i^j is introduced, where it is 1 if y_i belongs to the jth expert and 0 otherwise. The expected value of z_i^j, given the input x_i is

$$\begin{aligned} E[z_i^j|\mathcal{X}] &= f(z_i^j = 1|y_i, x_i, \theta^j) \\ &= \frac{f(y_i|z_i^j = 1, x_i;\theta^j) f(z_i^j = 1|x_i;\theta^j)}{f(y_i|x_i;\theta^j)} \\ &= \frac{f^j(y_i) g_i^j}{\sum_{r=1}^{J} g_i^r f_i^r(y_i)} \\ &= h_i^j \end{aligned} \tag{4.4.2}$$

The complete-data log likelihood is given by

$$
\begin{aligned}
l_c(\theta;\mathcal{Y}) &= \sum_{i=1}^{n}\sum_{j=1}^{J} z_i^j \ln\{g_i^j f^j(y_i)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{J} z_i^j \{\ln g_i^j + \ln f^j(y_i)\}
\end{aligned}
\tag{4.4.3}
$$

E step calculate the Q function as

$$
\begin{aligned}
l_c(\theta;\mathcal{Y}) &= \sum_{i=1}^{n}\sum_{j=1}^{J} h_i^j \{\ln g_i^j + \ln f^j(y_i)\} \\
&= Q_\gamma + Q_\theta
\end{aligned}
$$

Hence the M step consists of two separate maximisation problems. The updated estimate of $\gamma^{j(d+1)}$ is obtained by solving

$$
\sum_{i=1}^{n}\sum_{j=1}^{J} h_i^j \partial ln g_i^j(\gamma_j)/\partial\gamma = 0. \tag{4.4.4}
$$

The updated estimate of $\theta^{j(d+1)}$ is obtained by solving

$$
\sum_{i=1}^{n}\sum_{j=1}^{J} h_i^j \partial ln f_i^j(\theta^j)/\partial\theta = 0. \tag{4.4.5}
$$

Assuming the gating networks are logistic regressions as previously

$$g_i^j(x_i;\gamma^j) = \frac{\exp(\gamma^j x_i)}{\sum_{r=1}^{J}\exp(\gamma^r x_i)} \quad j = 1,\ldots,J \tag{4.4.6}$$

and hence (4.4.4) becomes

$$\sum_{i=1}^{n}\left(h_i^j - \frac{\exp(\gamma^j x_i)}{\sum_{r=1}^{J}\exp(\gamma^r x_i)}\right) \quad j = 1,\ldots,J \tag{4.4.7}$$

which is a set of nonlinear equations with $J \times p$ unknown parameters.

When the classification is a multiclass problem, j expert is just taken to be multinomial consisting of one draw on C categories. Hence the local output is modeled as

$$f^j(y_i = c|x_i;\theta) = \frac{\exp(\theta^{jc}x_i)}{\sum_{u=1}^{C}\sum\exp(\theta^{rc}x_i)}. \tag{4.4.8}$$

hence (4.4.5) becomes

$$\sum_{i=1}^{n} h_i^j\left(y_i - \frac{\exp(\theta^{jc}x_i)}{\sum_{u=1}^{C}\exp(\theta^{j}x_i)}.\exp(\gamma^r x_i)\right)x_i = 0 \quad u = 1,\ldots,C \tag{4.4.9}$$

for $j = 1,\ldots,J$ where are J sets of nonlinear equations each with $C \times p$ unknown parameters.

Looking at equation (4.4.4) the nonlinear expert of the jth expert not only depends on the parameters γ^j but also on other parameter vectors $\gamma^r(r = 1, \ldots, m)$. In other words each parameter β_j cannot be updated independently. With the Iteratively Reweighted Least Squares (IRLS) algorithm presented by Jordan and Jacob [1994], the independence assumption on those parameters was utilised explicitly and each parameter vector was updated independently and in parallel as

$$\gamma^{j(d+1)} = \gamma^{j(d)} + \rho_\gamma \left(\frac{\partial^2 Q_\gamma}{\partial \gamma^j \gamma^{jT}} \right) \frac{\partial Q_\gamma}{\partial \gamma_j} \quad (j = 1, \ldots, j = J) \tag{4.4.10}$$

where $\rho_\gamma < 1$ is the learning rate [Jordan and Xu, 1995]. Hence there are J set of nonlinear equations each with p variables instead of a set of a nonlinear equations with $J \times p$ variables. In Jordan and Jacob [1994] the iteration was termed as the inner loop of the of the EM algorithm.

Similarly for each parameter vector θ^{jc} for $(j = 1 \ldots, J)$ was updated independently as

$$\theta^{jc(d+1)} = \theta^{jc(d)} + \rho\gamma_\theta \left(\frac{\partial^2 Q_\theta}{\partial \theta^j{}_c \gamma^{jcT}} \right) \frac{\partial Q_\theta}{\partial \theta^j{}_C} (j = 1, \ldots, j = J) \tag{4.4.11}$$

where $\rho_\theta \leq 1$ is the learning rate. In the simulation experiment of Ng and McLachlan [2004] , ρ_γ and ρ_θ were set equal to 0.1. They adopted a smaller learning rate for x_i to ensure better convergence as y_i in (4.4.9) is binary zero

or one.

With reference to (4.4.10) and (4.4.11) the independence assumption on parameter vectors is equivalent to the adoption of an incomplete Hessian matrix of the Q function. Chen et al. [1999] proposed a learning algorithm based on the Newton Raphson Method for use in the inner loop of the EM algorithm. Particularly they pointed out that the parameter vectors cannot be updated separately due to the incorrect independence assumption. Rather they adopted the exact Hessian matrix in the inner loop of the EM algorithm. However, using the exact Hessian resulted in expensive computation during training. Hence they proposed a modified algorithm whereby approximate statistical model called the generalized Bernoulli density is introduced for expert networks in multiclass classification in that all of the off-diagonal block matrices in the Hessian matrices are zero matrices and so the parameter vectors θ^{jc} $(c = 1, \ldots, C)$ are separable. With this approximation, the learning time is decreased but the error rate increases [Chen et al., 1999].

Ng and McLachlan [2004] proposed an Expectation/Conditional Maximisation (ECM) algorithm for which both parameters vector γ^j and θ^{jc} are separable $j = 1, \ldots, J$ and $c = 1, \ldots, C$. The parameter γ is partitioned as $(\gamma^1, \ldots, \gamma^J)$. On the $d + 1$ iteration of the ECM algorithm, the E Step is the same as given for the above EM algorithm. On the M step the θ^{jc} are updated in one step. But the updating of γ is done over J conditional steps as follows:

- CM-Step 1: Calculate $\gamma^{1(d+1)}$ by maximising Q_γ with γ^r $(r = 2, \ldots, r =$

m) fixed at γ^l $(l = 2, \ldots, J)$ fixed at $\gamma^{r(d)}$

- CM-Step 2: Calculate $\gamma^{2(d+1)}$ by maximising Q_γ with γ^1 fixed at $\gamma^{1(d+1)}$ and γ^l $(r = 3, \ldots, J)$ fixed at $\gamma^{r(d)}$
  $\vdots$

- CM-Step m: Calculate $\gamma^J(d+1)$ by maximising Q_gamma with γ^r fixed at $\gamma^{r(d+1)}$ and γ^r $(r = 1, \ldots, m-1)$ fixed at $\gamma^{r(d)}$

As the CM maximisations are over a spaced of parameter space of fewer dimensions, they are usually simpler and more stable than the full maximisation called for in the M-step of the EM algorithm. More crucially each CM-step above corresponds to a separable set of the parameters in β_r for $r = 1, \ldots, J$ and can be obtained using the IRLS approach. This ECM algorithm preserves the appealing convergence properties of EM algorithm such as the monotone increasing of likelihood after each iteration.

In more recent work on mixture of experts, Ng et al. [2006] considered an incremental EM based algorithm in the context of online prediction of impatient length of stay (LOS). Ng and McLachlan [2007] have considered an extension of of mixture of experts networks for binary classification of correlated data with a hierarchical or clustered structure.

Evers [2007], tried another approach, instead of maximising the likelihood "indirectly" by applying the EM algorithm, the likelihood is maximised

directly. Modifications of second order methods like Newton's method might conceivably converge faster than the EM algorithm.

As the likelihood of HME models is not convex, the Newton's method cannot be blindly adopted as it it might not actually converge to a local maximum of the likelihood. Hence Newton's method has to be safeguarded. There are two schools of thought about how this can be done. Linear search methods (see e.g. Nocedal and Wright [2006],chapter 3) keep the direction proposed by Newton's method and look along this direction for a point yielding a sufficiently large improvement of the loglikelihood. The direction proposed by Newton's algorithm is not necessarily pointing uphill, so the search might have to be extended to the negative Newton direction. [Evers, 2007] chose to implement the other school of thought, which is base on the trust region approach (see e.g. [Nocedal and Wright, 2006],chapter 4). Trust region method, very much like the Newton's method, minimise a quadratic approximation to the log likelihood. They 'trust'the quadratic approximation only in region of radius ρ. If the proposed step does not yield a larger log likelihood then the proposed move is discarded and the radius ρ of the trust region is reduced. If the quadratic approximation and the likelihood are close, the the radius ρ is increased.

The trust region Newton method converges quadratically and the EM algorithm converges linearly so one might expect the Newton method to be the better approach. However there are two caveats. Firstly the quadratic convergence of the trust region method only applies when being close enough

to the solution. Secondly a single iteration of the EM algorithm is can be carried out faster than a single iteration of the trust-region. Evers [2007] did empirical tests and discoverd that the best approach is an hybird approach, whereby a couple of EM iterations are performed before switching over to direct optimization of the likelihood. Direct optimisation also avoids the problem of the independence assumption mentioned earlier. This"'mixed" approach has also be utilised in other statistical models as well. Pinheiro and Bates [2000] proposed using this approach for fitting linear mixed effect models.

All the only methods only ensures convergence to a local optimal. This local maximiser can be far from the global maximum. So we propose to use simulated annealing, a very popular global convergent algorithm. The weakness of such algorithms is that they can be very slow, especially if the parameter space is comparatively large as it can be for HME models. So we propose using a hybird approach like before. First we start off with the EM algorithm, then switch over to direct optimization and finally simulated annealing is used in the final phase. To enhance the effectiveness of the algorithm, we use ten different starting points and took the best one.

4.5 Determining the HME tree structure

The above discussion assumes that the HME structure is known a priori which is an unrealistic assumption. In reality, the HME structure will have

to be inferred in some way from the data. As determining the HME tree structure is not the focus of this thesis, we have chosen to use a two level HME with a binary split at each level resulting in four experts. We will nevertheless give a review of the commonly used approaches in which an HME tree can be determined.

The most direct and simple approach to model selection is to iteratively "grow" the HME model like classification trees. Starting of with a single expert, the model is grown by adding one (or more) experts in every step i.e. iteratively replacing one of the of the experts with a gate leading to two (or more) experts. The crucial part of this method is the choice of the expert to be split. The CART solution to this problem is by total enumeration: all possible splits are carried out and the one yielding the largest reduction of the cost function is chosen. In HME, this would correspond to fitting all possible models that can be obtained from an additional split and choosing the one resulting in the largest loglikelihood. As HME utilizes soft splits and the parameters are interdependent, all the model parameters would have to be retrained for every possible split. This is computationally very intensive and is only feasible for small number of parameters. Approximate methods would have to be used for larger problems.

Fritsch et al. [1997] suggested choosing the expert having the smallest path probability. This is not necessarily a wise strategy as the small likelihood can be due to one of three possible factors: firstly the model fit might be lousy in this area of this expert, there might be a lot of noise in this region

or the expert might only be specialised on a small number of observations (or even be "inactive"). Splitting the expert would only help in the first case.

Waterhouse and Robinson [1995] proposed an analogous method to CART models: to preserve all parameters except the newly introduced parameters and perform optimisation over the latter. This reduces the computation burden significantly. This algorithm, however only results in a lower bound on the improvement in the loglikelihood. The "harder" the relevant splits are i.e. the larger the norm of the relevant gating coefficients, the nearer the bound will be to the actual improvement. When the gating coefficients are penalised, the splits would be rather "soft" and hence the lower bound would be rather loose. Waterhouse [1997], reported that "Empirical investigation fail[ed] to show good performance of this approach". The reason for this might be specific to the way the parameters were estimated and initialised by Waterhouse. Waterhouse proposed an alternative method which consists of discretising the problem: the expert in question is replaced by a constant model and all possible hard splits are compared as in the CART algorithm. This approach however possess it own problems. A constant model (or largely simplified) regression model can be a very lousy approximation the expert. Then there is the question of how to proceed once the best split has been chosen, as it cannot represented in the HME model and hence the split has to be artificially "softened".

Evers [2007] proposed an algorithm based on dispersion score tests. As they are score tests, they all can be carried out without having to compute

the loglikelihood under the alternative model featuring the additional split and hence are considerably faster than both approaches proposed by Waterhouse. In addition, they correspond to tests which have known asymptotic distribution.

4.6 Regularisation Techniques

4.6.1 Introduction

The big disadvantage of maximum likelihood estimation is that it usually leads to overfitting. There are two possible ways around it. One way, which would be described in the next chapter is to adopt the Bayesian approach. The other way, which we are going to describe below is via regularisation.

4.6.2 Early Stopping

An ad-hoc regularsiation technique that was frequently used in training neural networks is early stopping.

The basic premise behind early stopping is not to carry out the entire optimisation but to abort the optimisation process before the parameters have fully converged to their maximum-likelihood estimates. The rationale is that "early" parameters have better generalisation performance. One would not know beforehand when is the optimal time to stop in order to obtain the the best generalization performance. What can be done is to compute

the predicted likelihood on a validation dataset and select the stopping time that produces the largest likelihood on the validation set i.e. the minimum generalisation error.

In early stopping, every iteration is considered a potential model and not just the model obtained after convergence. If we assume that that overfitting only occurs at the the end of the optimisation process, the sequence of models generated by the early stopping algorithm can be interpreted as a sequence starting from a strongly regularised model to a completely un-regularlised model.

However it is pretty unrealistic to assume that we can stop performing optimisation process before overfitting commences, especially for HME models as they have huge number of parameters. To be able to do that successfully relies on the assumption that overfitting occurs concurrently for all the parameters. It can be very well that some parameters affecting one part of the covariate space are already overfitted whereas the parameters effecting another part of the covariate space are far from being optimal.

4.6.3 Penalised Likelihood

The two most commonly used methods used in penalised likelihood estimation are lasso or ridge regression. We are just going to focus on using ridge regression here as it is equivalent to adopting a normal prior on the regression parameters when we adopt a Bayesian approach in the next chapter. In ridge regression, what happens is that we add a penalty for the l_2 norm

of regression coefficients. We will consider penalising both the gating and expert coefficients separately.

4.6.3.1 Penalizing the gating networks

The penalty is of the form:

$$\lambda \sum_{j,k} \| \gamma^{jk} \|^2 .$$

By penalizing the norm of the gating coefficients γ^{jk} and γ^{j} we ensure that they will not take too large values, and hence yields "softer" transitions between the experts. These softer transitions result in a loglikelihood with fewer local extrema, thus simplifying the optimisation process.

Furthermore this penalty function ensures that even if there is perfect separation between two experts, the loglikelihood is bounded in the gating parameters i.e. the gating parameters would not diverge to $\pm\infty$.

4.6.3.2 Penalizing the experts

Another way of penalizing the likelihood function is to introduce an quadratic penalty for the expert parameters of the form:

$$\lambda \sum_{j,k} \| \beta^{jk} \|^2 .$$

In some situations, this can prove useful. In the instance where a large number of covariates are used, penalizing both the expert parameters and the gating parameters can produce a significant improvement in the generalisation performance and since our experts are logit models, penalizing the expert parameters ensures that the expert parameters are bounded from above. Since both the gating and expert networks are both logistic regression models, it can be that in certain parts of the covariate space, all the variation is already explained by the gating networks, thus resulting in the unregularised loglikelihood being unbounded in some of the expert parameters.

4.6.4 Choosing the penalisation value

Within the maximum likelihood paradigm, the most common approach for choosing the penalty values is a grid like approach of permuting all the possible likely values of λ and choosing the set of values that results in the lowest cross-validated error rate. That still leaves us with the issue of which penalty values to permute with. Ripley [1996] suggests taking a Bayesian perspective [Buntine and Weigend, 1991, Ripley, 1994]. Suppose E is the negative log-likelihood, up to a constant or half the deviance. Then if we take a prior distribution for the parameters with density $f(\beta) = \exp -\lambda \sum \beta^2$ the minimiser of $E + \lambda \sum \beta^2$ will maximize the posterior density of the weights.

This prior corresponds to a Gaussian distribution with mean zero and variance $1/2\lambda$. Since the logistic function saturates for inputs beyond around ± 3, we might expect the standard deviation of the total input to be around

2 (one motivation for using penalisation is to avoid unnecessary saturation of logistic units). If the inputs are scaled to between $[0, 1]$, this would suggest that the standard deviation of the weights would be around 5, which would correspond to $\lambda = 1/50$. This is a rather conservative argument as we would want some of the weights to saturate. A range of $\lambda \approx 0.001 - 0.1$ would seem reasonable. Previous experience shows that choice of λ is not critical within a factor of 5.

4.7 Results

4.7.1 Model Selection

The penalisation value controls the amount of regularisation so choosing a good value is important. Because each penalisation value corresponds to a fitted model, choosing a penalisation value essentially corresponds to model selection. What is considered a good choice of penalisation value depends our objectives, to optimise prediction accuracy or recover the "right model" for interpretation purposes. We will focus on the former since our aim is to compare prediction accuracy of the frequentist approach to the Bayesian approach.

Predictive accuracy was assessed using fivefold cross validation and we choose the value of the penalisation value that gives the minimises the cross-validated error rate. The error rate (percent) and standard error for the all

the data sets are given below.

4.7.2 Crabs Data

Table 4.1: Cross Validated Results for Crabs Data

$g1$	$g2$	ex	Error	SD
0.001	0.001	0.001	28.4	1.4
0.001	0.001	0.01	28.7	1.5
0.001	0.001	0.1	29.8	1.2
0.001	0.01	0.001	28.4	1.4
0.001	0.01	0.01	28.7	1.5
0.001	0.01	0.1	29.8	1.2
0.001	0.1	0.001	28.4	1.4*
0.001	0.1	0.01	28.7	1.5
0.001	0.1	0.1	29.8	1.2
0.01	0.001	0.001	28.7	1.5
0.01	0.001	0.01	28.7	1.5
0.01	0.001	0.1	29.8	1.2
0.01	0.01	0.001	28.7	1.5
0.01	0.01	0.01	28.7	1.5
0.01	0.01	0.1	28.7	1.5
0.01	0.1	0.001	28.7	1.4
0.01	0.1	0.01	28.7	1.5
0.01	0.1	0.1	29.7	1.2
0.1	0.001	0.001	29.0	1.3
0.1	0.001	0.01	28.7	1.5
0.1	0.001	0.1	29.7	1.2
0.1	0.01	0.001	29.0	1.3
0.1	0.01	0.01	28.7	1.5
0.1	0.01	0.1	29.7	1.2
0.1	0.1	0.001	29.0	1.3
0.1	0.1	0.01	28.7	1.5
0.1	0.1	0.1	29.7	1.2

As expected, the penalisation value of the experts has the heaviest influence on the error rate, with smaller penalisation values generally giving lower accuracy rates. This holds true only at lower penalisation values of the gating networks. At larger penalisation values for the gating networks, intermediate

values of penalisation values of the experts gives the best results. There is a tie between two models for the lowest error rate and we break the tie by choosing the simpler model (more regularised) model.

4.7.3 Music Data

Table 4.2: Cross Validated Results for Music Data

$g1$	g2	ex	Mean	Sd
0.001	0.001	0.001	19.1	1.4
0.001	0.001	0.01	19.1	1.0
0.001	0.001	0.1	20.2	1.3
0.001	0.01	0.001	19.1	1.4
0.001	0.01	0.01	19.1	1.0
0.001	0.01	0.1	20.2	1.3
0.001	0.1	0.001	18.9	1.5*
0.001	0.1	0.01	19.1	1.0
0.001	0.1	0.1	20.2	1.4
0.01	0.001	0.001	19.0	1.4
0.01	0.001	0.01	19.1	1.0
0.01	0.001	0.1	20.2	1.3
0.01	0.01	0.001	19.0	1.4
0.01	0.01	0.01	19.1	1.1
0.01	0.01	0.1	20.2	1.4
0.01	0.1	0.001	19.0	1.4
0.01	0.1	0.01	19.1	1.0
0.01	0.1	0.1	20.2	1.3
0.1	0.001	0.001	19.2	1.5
0.1	0.001	0.01	19.2	1.6
0.1	0.001	0.1	20.2	1.3
0.1	0.01	0.001	19.2	1.5
0.1	0.01	0.01	19.2	1.1
0.1	0.01	0.1	20.2	1.3
0.1	0.1	0.001	19.1	1.6
0.1	0.1	0.01	19.2	1.1
0.1	0.1	0.1	20.2	1.3

We observe that the smallest penalisation value for the experts give the best results and the penalisation values of the gating networks does not affect the

prediction accuracy significantly.

4.7.4 Crystals Data

Table 4.3: Cross Validated Results for Crystals Data

g1	g2	ex	Mean	Sd
0.001	0.001	0.001	19.3	0.7
0.001	0.001	0.01	19.1	0.7
0.001	0.001	0.1	19.4	0.7
0.001	0.01	0.001	19.3	0.7
0.001	0.01	0.01	19.1	0.7
0.001	0.01	0.1	19.4	0.7
0.001	0.1	0.001	19.2	0.7
0.001	0.1	0.01	19.1	0.7
0.001	0.1	0.1	19.4	0.7
0.01	0.001	0.001	19.4	0.7
0.01	0.001	0.01	19.0	0.7
0.01	0.001	0.1	19.4	0.7
0.01	0.01	0.001	19.3	0.7
0.01	0.01	0.01	19.0	0.7
0.01	0.01	0.1	19.4	0.7
0.01	0.1	0.001	19.3	0.7
0.01	0.1	0.01	19.0	0.7
0.01	0.1	0.1	19.5	0.7
0.1	0.001	0.001	19.4	0.7
0.1	0.001	0.01	21.5	0.5
0.1	0.001	0.1	19.4	0.6
0.1	0.01	0.001	19.4	0.8
0.1	0.01	0.01	18.9	0.7*
0.1	0.01	0.1	19.5	0.6
0.1	0.1	0.001	19.4	0.8
0.1	0.1	0.01	19.0	0.7
0.1	0.1	0.1	19.5	0.6

We observe that moderate values of the penalisation values for experts work gives the best results. For the non optimal penalisation value of expert networks, increasing the penalisation value of the gating networks generally worsens the error rate whereas at the optimal value, increasing the penalisa-

tion value generally improves the error rate slightly.

4.7.5 Forensic Glass Data

Table 4.4: Cross Validated Results for Forsenic Glass Data

g1	g2	ex	Mean	Sd
0.001	0.001	0.001	25.2	2.0*
0.001	0.001	0.01	26.7	2.8
0.001	0.001	0.1	28.0	1.5
0.001	0.01	0.001	25.2	2.0
0.001	0.01	0.01	26.7	2.0
0.001	0.01	0.1	28.0	1.5
0.001	0.1	0.001	25.2	2.0
0.001	0.1	0.01	26.7	2.0
0.001	0.1	0.1	28.0	1.5
0.01	0.001	0.001	25.2	2.0
0.01	0.001	0.01	27.1	2.0
0.01	0.001	0.1	28.0	1.5
0.01	0.01	0.001	25.2	0.2
0.01	0.01	0.01	27.1	1.8
0.01	0.01	0.1	28.0	1.5
0.01	0.1	0.001	25.2	2.0
0.01	0.1	0.01	26.5	1.7
0.01	0.1	0.1	28.2	1.3
0.1	0.001	0.001	25.2	2.3
0.1	0.001	0.01	26.5	1.7
0.1	0.001	0.1	28.2	1.4
0.1	0.01	0.001	25.2	2.2
0.1	0.01	0.01	26.5	1.7
0.1	0.01	0.1	28.2	1.3
0.1	0.1	0.001	30.8	1.2
0.1	0.1	0.01	31.3	0.1
0.1	0.1	0.1	31.5	0.1

This time around the error rate increases dramatically as the penalisation values increases and the best prediction accuracy occurs at the smallest penalisation values for both the gating and expert networks.

4.7.6 WaveForm Data

Table 4.5: Cross Validated Results for WaveForm Data

g1	g2	ex	Mean	SD
0.001	0.001	0.001	12.9	0.3
0.001	0.001	0.01	13.0	0.3
0.001	0.001	0.1	12.8	0.3
0.001	0.01	0.001	12.9	0.3
0.001	0.01	0.01	13.0	0.3
0.001	0.01	0.1	12.8	0.3
0.001	0.1	0.001	12.9	0.3
0.001	0.1	0.01	13.0	0.3
0.001	0.1	0.1	12.8	0.3
0.01	0.001	0.001	12.9	0.3
0.01	0.001	0.01	13.0	0.3
0.01	0.001	0.1	12.8	0.3
0.01	0.01	0.001	12.9	0.3
0.01	0.01	0.01	13.0	0.3
0.01	0.01	0.1	12.8	0.3
0.01	0.1	0.001	12.9	0.3
0.01	0.1	0.01	13.0	0.3
0.01	0.1	0.1	12.8	0.3
0.1	0.001	0.001	13.0	0.3
0.1	0.001	0.01	12.9	0.3
0.1	0.001	0.1	12.8	0.3
0.1	0.01	0.001	13.0	0.3
0.1	0.01	0.01	13.0	0.3
0.1	0.01	0.1	12.8	0.3
0.1	0.1	0.001	13.0	0.1
0.1	0.1	0.01	12.9	0.7
0.1	0.1	0.1	12.8	0.7*

This time around we observe that heavily penalising the experts give the best error rates and the penalisation values of the gating networks do not affect the error rate and hence we chose the model with the largest penalisation values for both the gating networks and experts

4.7.7 ZipCode Data

Table 4.6: Cross Validated Results for Zip Code Data

g1	g2	ex	Mean	SD
0.001	0.001	0.001	4.49	0.18
0.001	0.001	0.01	4.23	0.17
0.001	0.001	0.1	3.56	0.19
0.001	0.01	0.001	4.45	0.19
0.001	0.01	0.01	4.20	0.16
0.001	0.01	0.1	3.56	0.19
0.001	0.1	0.001	4.38	0.18
0.001	0.1	0.01	4.34	0.16
0.001	0.1	0.1	3.56	0.19
0.01	0.001	0.001	4.45	0.19
0.01	0.001	0.01	4.23	0.13
0.01	0.001	0.1	3.54	0.18
0.01	0.01	0.001	4.45	0.19
0.01	0.01	0.01	4.25	0.15
0.01	0.01	0.1	3.54	0.20
0.01	0.1	0.001	4.38	0.18
0.01	0.1	0.01	4.20	0.17
0.01	0.1	0.1	3.56	0.15
0.1	0.001	0.001	4.45	0.18
0.1	0.001	0.01	4.23	0.13
0.1	0.001	0.1	3.56	0.19
0.1	0.01	0.001	4.43	0.16
0.1	0.01	0.01	4.23	0.18
0.1	0.01	0.1	3.54	0.20*
0.1	0.1	0.001	4.45	0.22
0.1	0.1	0.01	4.16	0.15
0.1	0.1	0.1	3.56	0.19

Again we observe that heavy penalisation of the experts gives the best results. For gating 2 level network, accuracy improves initially as the penalisation value increases but decreases as it increases more. For gating 1 level network, the accuracy improves as we move towards heavier penalisation.

4.8 Comparing with other machine learning models

To compare our results with other classification algorithms, we use a statistical test for comparing machine learning algorithms developed by Dietterich [1998]. We will describe briefly the test here. In this test, five replications of two-fold cross-validation is performed. In each replication, the available data are randomly partitioned into two equal size sets S_1 and S_2. Each learning algorithm (A and B) is trained on each set and tested on the other set. This produces four error estimates $p_A^{(1)}$ and $p_B^{(1)}$ (trained on S_1 and tested on S_2), $p_A^{(2)}$ and $p_B^{(2)}$ (trained on S_2 and tested on S_1) where p is the proportion of test examples incorrectly classified by the machine learning algorithm. Subtracting corresponding error estimates gives us two estimated differences $p^{(1)} = p_A^{(1)} - p_B^{(1)}$ and $p^{(2)} = p_A^{(2)} - p_B^{(2)}$. From these two differences, the estimated variance is $s^2 = (p^{(1)} - \bar{p})^2 + (p^{(2)} - \bar{p})^2$ where $\bar{p} = (p^{(1)} + p^{(2)})/2$. Let s_i^2 be the variance computed from the i-th replication and let $p_1^{(1)}$ be the $p^{(1)}$ from the very first of the five replications. Then define following statistic

$$\tilde{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}} \tag{4.8.1}$$

which they termed as the 5×2cv $\tilde{t}$ statistic. Under the null hypothesis, it has approximately a t distribution with five degrees of freedom. For full details

please see Dietterich [1998]

We compare our method with the most commonly used classification algorithms, random forests, classification trees, support vector machines, neural networks mixture of normals, classification trees. The alternative machine learning algorithms were tuned using the Caret package in R [Kuhn, 2012]

Table 4.7: Comparing cross-validated results with other models

Model/Data Set	**Crabs**	**Music**	**Crystals**	**Glass**	**Wave**	**Zip**
Neural Nets	30.8 (5.8)	23.6 (5.9)	20.4 (3.4)	29.1 (0.8)	13.0 (3.7)	6.7(4.0)
Random Forest	6.0 (7.9)	25.3 (5.6)	20.7 (4.2)	21.3 (6.4)	13.9 (2.6)	4.5(3.6)
C5.0	35.3 (7.9)	25.8 (5.6)	22.1(5.5)	23.0(7.2)	15.6 (3.0)	4.0 (3.4)
SVM-Poly	34.5 (7.1)	22.3 (6.3)	20.2 (4.5)	31.5 (6.7)	12.8 (3.0)	2.0 (2.7)
SVM-Rad	31.0 (8.4)	25.9 (6.5)	21.0 (3.7)	34.4 (7.7)	12.6 (3.2)	3.3 (2.8)
MDA	39.3 (1.5)	29.1 (8.3)	23.0 (4.7)	33.5 (7.8)	13.2 (3.1)	6.5 (3.2)
ME-Freq	28.7 (1.5)	18.9 (1.3)	18.9 (7.2)	25.2 (2.2)	12.8 (0.2)	3.5(3.2)

We perform the 5×2cv test for our model against the alternative methods and the results are given below. A $+$ means our model outperformed the alternative, a $-$ means the alternative outperformed our model and a 0 indicates shows there is insufficient evidence to show which model is significantly better.

Table 4.8: Comparisons with other models using a statistical test

Model/Data Set	**Crabs**	**Music**	**Crystals**	**Glass**	**Wave**	**Zip**
Neural Nets	+	+	+	+	+	+
Random Forest	+	+	+	-	+	+
C5.0	+	+	+	-	+	+
SVM-Poly	+	+	+	+	0	-
SVM-Rad	+	+	+	+	-	0
MDA	+	+	+	+	+	+
ME-Freq	+	+	+	+	+	+

From the results we see that our model has done reasonably well, outperforming most of the alternative models on most data sets, except for random forest and trees on the glass data set, support vector machines on the waveform and zip code data set. And in two of these cases, support vector machines with polynomial kernel on the waveform dataset, support vector machines with radial kernel on the zip code data set ,there is insufficient evidence to show that the alternative algorithm is better, even if we cannot show that our chosen model works better.

For the forsenic glass data set, there are small numbers in each class logistic regression is well known to perform badly in this instance: small training data in a highly parameterised model. For the waveform data set, due to the complex way the subclasses and the features are generated and the complex nature of the mixture of experts, makes finding a good optimum

difficult. The Zip code data set is a very high dimensional problem which in conjunction with the complex nature of mixture of experts, makes finding a good optimum difficult.

Despite the weaknesses of our model in this situations, we still chose this model as it allows us to incorporate the subclass features. And also, Bayesian methods have been demonstrated to work well for smoothly parameterised models like neural networks, something which has yet to be demonstrated for trees and SVMs. Our model has outperformed all other smoothly parmeterised models and being a simpler model than neural networks, makse the implementation of Monte Carlo Methods much easier, which will be the focus of next chapter.

Chapter 5

Bayesian fitting of Mixture of Experts

5.1 Introduction

If mixture of experts are not popular because fitting by maximum likelihood estimation is difficult, Bayesian treatments are even rarer. Most of the early attempts at Bayesian treatment of mixture of experts relied on approximation methods like Laplace or variational inference. There has been attempts to obtain its posterior density by MCMC [Jacobs et al., 1996, Fengchun et al., 1996, Jacobs et al., 1997, Villagran and Huerta, 2005] but it is not common. And this is what are we are going to attempt to do in this thesis, to the obtain the posterior density of the parameters needed for the predictive approach by MCMC simulation.

5.2 Literature Review

MCMC approaches which are relevant towards simulation of mixture of experts can be loosely classified into three camps: neural networks, mixture of experts, and generalized linear models/logistic regression and we will review the research in that order.

Neal [1996] introduced an MCMC implementation of Bayesian learning for neural network based on the Hybrid Monte Carlo Scheme (HMC) [Duane et al., 1987] which merges the Metropolis-Hastings algorithm with sampling techniques based on dynamic simulation. Adoption of HMC in the statistical community is in general not widespread because it is extremely difficult to implement. In recent years, there has been a revival of interest in the use of HMC. Ishwaran [1999] applied the HMC to Bayesian logistic regression with quasicomplete separation with excellent results and more recently Girolami and Calderhead [2011], proposed a Riemann manifold Langevin and Hamiltonian Monte Carlo sampler to resolve the shortcomings of existing Monte Carlo algorithms when sampling from densities that maybe be high dimensional and exhibit strong correlations. The algorithm displayed excellent results when applied to logistic regression.

The earliest and to date known (to the author) attempts [Jacobs et al., 1996, Fengchun et al., 1996, Jacobs et al., 1997, Villagran and Huerta, 2005] to simulate mixture of experts utilises the plain vanilla random walk Metropolis Hastings algorithm.

Early attempts to simulate Bayesian logistic regression under the umbrella of generalised linear models include Albert and Chib [1993], Dellaportas and Smith [1993], Gamerman [1997], Chen and Dey [1998]. Della and Smith used a Gibbs sampling scheme which is based on the adaptive rejection sampling technique [Gilks and Wild, 1992]. The weakness of this approach is that it is an univariate technique and hence can be potentially very inefficient if the parameters are high correlated, which is frequently the case. Chen and Day based their logistic regression on a scale mixture representation. Their approach requires the evaluation of the mixing density, which is known as an infinite series expansion, necessitating the use of approximate numerical techniques. Albert and Chib [1993] used an auxiliary variable approach for binary probit regression that reduces conditional likelihood of the model parameters to one that is equivalent to those under the Bayesian normal linear regression model with Gaussian noise. Hence conjugate priors are available and the block Gibbs sampler can be utilized to great effect. Holmes and Held [2006] showed that this auxiliary approach is possible for logistic regression, by representing the noise process with a scale mixture of normal distributions and several other papers Fruhwirth-Schnatter and Fruhwirth [2007], Obrien and Dunson [2004], Scott [2009] were written in similar vein. Recently Polson et al. [2012] came up with another conjugate prior/auxiliary variable approach based on Póly-Gamma latent variables.

The major drawback of these approaches are that they require the use of a conjugate prior (more specifically the normal prior). While in this case

it so happens that the conjugate prior utilised is the normal prior which corresponds to the prior that we are using here, we would like to seek a more general MCMC method which does not constrain us in the choice of priors.

Hence the method we chose initially to implement was Gamerman's method which uses a Metropolis-Hastings algorithm, which attempts to overcome the difficulties mentioned in the methods described earlier by incorporating the structure of the model.

5.3 Gamerman's Algorithm

5.3.1 Review of Markov Chain Monte Carlo methodology

Suppose we have a distribution (identified with its density) π on a random quantity x blocked into components $x_1, \ldots x_m,$, that can themselves be vectors or matrices. In its most basic form, MCMC methodology is based on constructing a transition density $q(x, x^*)$ so that its Markov chain has equilibrium probability given by π. A draw x generated from π is obtained by

1. Start with $x = x^0$ and set d=1;

2. Sample x^d from $q(x^{d-1}, x^d)$;

3. Increase d by 1 and and return to step 2.

For large enough d, X^d will be a draw from π for any practical purpose.

A sample from π is generated by retaining sufficient values from the chain after convergence is reached. Approximate independence between draws is achieved keeping only every m generated value to break chain correlations.

One such scheme is Gibbs sampling. Its transition is formed by successively sampling from the full conditional densities $\pi(x_k) = \pi(x_k|x_{-k})$ where $x_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, ldots, x_m),\ k = 1, \ldots, m$. For many problems, this schemes works well but in our case the $\pi_k's$ are hard to sample from.

Another algorithm is the Metropolis-Hasting algorithm. Consider a general transition density $q(x, x^*)$ and define

$$\alpha(x, x^*) = min \left\{ 1, \frac{\pi(x^*)q(x^*, x)}{\pi(x)q(x, x^*)} \right\}.$$

The move from state x^{d-1} to x^d is made as follows: (a) sample x^* from $q(x^{d-1}, x)$; (b) accept the move to x^* with probability $\alpha(x^{d-1}, x^*)$ and set $x^d = x^*$. Otherwise stay at $x^d = x^{d-1}$. Because of the acceptance stage (b) q is usually called the proposal (transition) density.

5.3.2 Review of Generalised Linear Models

Following the notation of McCullagh and Nelder [1989], the data set consists of n observations with univariate response y_i and a p dimensional vector of covariates x_i $i = 1, \ldots, n$. The observations are assumed to be independent

with exponential family density:

$$f(y_i|\theta_i) = \exp[(y_i\theta_i - b(\theta_i))/\phi_i]c(y_i, \phi_i). \tag{5.3.1}$$

The means $\mu_i = E(y_i|\theta_i)$ are linked to the canonical parameters θ_i via $\mu_i = b'(\theta_i)$ and to the other regression coefficients β by the link function:

$$g(\mu_i) = \eta_i = x_i\beta \qquad i = 1, \ldots, n. \tag{5.3.2}$$

5.3.3 Iteratively weighted least squares (IWLS)

The maximum likelihood (ML) estimator in a GLM and its asymptotic variance can be obtained by iterative use of weighted least squares (WLS) to transformed observations. The algorithm for the IWLS starts off by transforming the observations y_i to $\tilde{y}_i$ by:

$$\tilde{y}_i(\beta) = \eta_i + (y_i - \mu_i)g'(\mu_i). \tag{5.3.3}$$

The associated diagonal matrix of weights W_i is given by:

$$W_i^{-1}(\beta) = b''(\theta_i)\{g'(\mu_i)\}^2 \tag{5.3.4}$$

The iterative weighted least squares algorithm is as follows:

1. Start with $\beta = m^0$ and set d=1;

2. Obtain m^d, the WLS estimator of β as if $\tilde{y}(m^{d-1}) \sim N(X\beta, W^{-1}(m^{d-1}))$ and its associated covariance matrix C^d;

3. Increase d by 1 and return to the step 2,

where

$$m^d - (X^T W(m^{d-1})X)^{-1} X^T W(m^{d-1})\tilde{y}(m^{d-1})$$
$$C^d - (X^T W(m^{d-1})X)^{-1}.$$

Note that both of them are functions of the previous m^{d-1}.

The detailed GLM formulation for IWLS is given in Appendix A.

5.3.4 Bayesian version of IWLs

The Bayesian version of IWLs algorithm, developed by West [1985], for the special case of the canonical link $\theta_i = \eta_i$, but it is relatively straightforward to extend it for general link functions. It gives us a posterior mode and an approximate posterior covariance matrix for β. The main idea is to combine step 2 of the IWLS with a N(a,R) prior of β with "observations" $\tilde{y}(m^{(d-1)}) \sim N(X\beta, W^{-1}(m^{(d-1)}))$ Values of and m^d and C^d are now:

$$m^d - (R^{-1} + X^T W(m^{d-1})X)^{-1}\{R^{-1}a + X^T W(m^{d-1})\tilde{y}(m^{d-1})\}$$
$$C^d - (R^{-1} + X^T W(m^{d-1})X)^{-1}$$

If the prior is non-informative ($R^{-1} \to 0$), we recover the original IWLS algorithm. Both approaches utilise the IWLS algorithm combined with asymptotic results to make inferences based on approximate normality. Based on this, Gammerman proposed a MCMC algorithm that retains the structure of the IWLS without any necessity of resorting to any possible normality assumptions that might prove inadequate.

5.3.5 A weighted least squares proposal

The iterative schemes elaborated in Sections (5.3.1) and Sections (5.3.3) are very similar in nature and it seems natural to combine them in a single iteration cycle. The crucial steps in the cycles are Steps 2 for which the MCMC cycle requires a value proposed from a proposal density and the Bayesian IWLS provides such a density but does not sample from it. A single iterative cycle that combines these two is as follow:

1. Start with $\beta = m^0$ and set d=1;

2. Calculate m^d and C^d;

3. Sample β^* from $N(m^d, C^d)$;

4. Accept the move with probability $\alpha(\beta^d, \beta^*)$ and set $\beta^d = \beta^*$. Otherwise, stay at $\beta^d = \beta^{d-1}$;

5. Increase d by 1 and return to step 2.

And the moments of the proposal density m^d and C^d are as before:

$$m^d - (R^{-1} + X^T W(m^{d-1})X)^{-1}\{R^{-1}a + X^T W(m^{d-1})\tilde{y}(m^{d-1})\}$$
$$C^d - (R^{-1} + X^T W(m^{d-1})X)^{-1}$$

so that the transition is made from the previous state β^{d-1} and the acceptance probability is given by:

$$\alpha(\beta^d, \beta^*) = \frac{\pi(x^*)q(\beta^*, \beta^{d-1})}{\pi(x^{d-1})q(\beta^{d-1}, \beta^*)}, \tag{5.3.5}$$

where π is the stationary distribution/posterior density of β and $q(\beta^*, \beta^d - 1)$ is a $\mathcal{N}(M^*, C^*)$ density evaluated at β^{d-1} and m^* and C^* have the same expression as m^d and C^d but depend on β^* instead of β^{d-1}.

The main disadvantage of this method is that the moments of the proposal distribution has to be recalculated at every iteration. However in Gamerman [1997] this method demonstrated a high acceptance rate of the proposed moves e.g. 98%.

5.3.6 Gamerman's Algorithm applied to HME

Mixture of experts are not GLMs but with the use of indicator variables as in the previous chapter , we can formulate the likelihood as blocks of GLM. As mentioned in the previous chapter, we are going to impose a normal prior on the unknown parameters. The prior distribution is identical to the weight decay prior commonly used in neural networks. Formally, the prior is of the

form $p(\beta) = \exp -\lambda \sum \beta^2$, where λ is to be determined. And as alluded previously, imposing such a prior is equivalent to penalising the likelihood with the quadratic penalty term we have introduced in the previous chapter. Combining the prior with our likelihood, our posterior density (with the indicator variables) takes the following form:

$$\left[\prod_{j=1}^{J} f(\gamma^j) \prod_{j=1}^{J} \prod_{k=1}^{K_j} f(\gamma^{jk}) \prod_{j=1}^{J} \prod_{k=1}^{K_j} f(\theta^{jk}) \prod_{i=1}^{n} \prod_{j=1}^{J} \prod_{k=1}^{K_j} g_i^j g_i^{k|j} f^{jk}(y_i) \right]^{z_i^{jk}}, \quad (5.3.6)$$

where

$$f(\gamma^j) = \exp -\lambda_1 \sum (\gamma^j)^2$$
$$f(\gamma^{jk}) = \exp -\lambda_2 \sum (\gamma^{jk})^2$$
$$f(\theta^{jk}) = \exp -\lambda_3 \sum (\theta^{jk})^2.$$

Unlike the EM algorithm in maximum likelihood framework where the $z_i^{jk}s$ are imputed with its expected values, they are simulated with the value 1 or 0 based on its distribution (Equation (**??**)) and given the value of $z_i^{jk}s$, the posterior density can be decomposed into 3 separate blocks of GLM/logistic regression with a prior for each logistic regression (which is equivalent to penalising its loglikelihood with a quadratic penalty as mentioned in the

previous chapter), with each block corresponding to each network:

$$\left[\prod_{i=1}^{n}\prod_{j=1}^{J} f(\gamma^j)g_i^j\right]\left[\prod_{i=1}^{n}\prod_{j=1}^{J}\prod_{k=1}^{K_j} f(\gamma^{jk})g_i^{k|j}\right]\left[\prod_{i=1}^{n}\prod_{j=1}^{J}\prod_{k=1}^{K_j} f(\theta^{jk})f^{jk}(y_i)\right] \quad (5.3.7)$$

Hence Gamerman's method can be applied as part of a blocked Metropolis Hastings Algorithm with each network as a separate independent block.

5.3.7 Attempts to Improve our MCMC Algorithm

5.3.7.1 Line Search Method

Gamerman's algorithm only worked when the dimensions of the problem is low, the acceptance rates being unacceptably low when the dimensions are high. One reason which we discovered was that the proposal was proposing moves that were too large.

The iterative weighted least squares algorithm arose out of applying the Fisher's scoring algorithm to the loglikelihood [McCullagh and Nelder, 1989]. Fisher's scoring is a form of the Newton-Rapson method. For the canonical link in GLMs, as it is in the case of logistic regression, Fisher's scoring and Newton-Rapson method are identical and henceforth we would use the two methods interchangeably. The Newton-Rapson method can be interpreted as a version of the line search method.

In each iteration of the line search method, a search direction p_k is computed and then decides how far to move along that direction. The iteration

is given by:

$$x_k + \alpha_k p_k.$$

In the Newton-Rapson method, the search direction p_k is the Newton's direction $-(f''(x))^{-1}f'(x)$ and the step length α_k is unity. When Gammerman's method was first applied to our problem, it did not work well. Acceptance rates of moves were close to zero. Upon investigation, it was discovered that the reason for that was that unit step length implicit in the Newton-Rapson method was too large a move. Hence we implemented a line search method preserving the Newton direction as the search direction and a step length which is much less than unity.

This however did not improve the mixing rate of our MCMC chain significantly.

5.3.7.2 Adaptive MCMC methods

One common reason the Metropolis Hasting algorithm does not work well is that the posterior distribution is a poor match for the actual distribution. Adaptive MCMC methods attempt to solve this problem by allowing the proposals to depend on the previous values of the chain. Haario et al. [2001] provided a a scheme that is guaranteed to converge to the correct distribution. This approach has become an increasing popular line of research and current ideas are reviewed by Andrieu and Thoms [2008] . We tried

implementing a version of the original adaptive MCMC developed by Haario. The algorithm is as follows. We begin with an a d-dimension algorithm target distribution $\pi(.)$ with proposal distribution given at iteration n, $Q_n(x,.) = \mathcal{N}(x,(0.1)^2 I_d/d$ for $n \leq 2d$, while for $n > 2d$

$$Qn(x,.) = (1-c)\mathcal{N}(x,(2.38)^2\Sigma_n/d) + c\mathcal{N}(x,(0.1)^2 I_d/d) \tag{5.3.8}$$

where Σ is the current empirical estimate of the covariance structure of the target distribution based on the run so far and c is a small positive constant (we take $c = 0.05$). It however did not improve the mixing rate of our algorithm significantly.

5.4 MCMC inference of Logistic Regression using Holmes and Held Auxiliary Variable Model

Despite our best attempts to improve it, Gamerman's method did not work very well, the MCMC chain taking too long to converge. We decided that the use of a conjugate prior was necessary and we chose to auxiliary variable method developed by Holmes and Held [2006], henceforth $H\&H$ which at the time of writing this thesis was the only conjugate MCMC algorithm available for logistic regression

5.5 Data augmentation in binary regression models

To begin with, consider the Bayesian binary regression model,

$$\begin{aligned} y_i &\sim Bernoulli(g^{-1}(\eta_i)) \\ \eta_i &= x_i\beta \\ \beta &\sim \pi(\beta) \end{aligned} \tag{5.5.1}$$

where $y_i \in \{0, 1\}, i = 1, \ldots, n$ is a binary response variable for a collection of n objects with associated p features $\mathbf{x} = (x_{i1}, \ldots x_{ip})$, $g(\mu)$ is the link function, η_i denotes the linear predictor and β represents a $(p \times 1)$ column vector of regression coefficients which *a priori* are from some distribution $\pi(\cdot)$.

5.6 Probit regression using auxiliary variables

For the probit link $g^{-1}(u) = \Phi(u)$ where $\Phi(u)$ represents the cumulative distribution function of a standard normal random variable. The model in (5.5.1) has a well known representation using auxiliary variables.

$$
\begin{aligned}
y_i &= \begin{cases} 1 \text{ if } z_i \geq 0 \\ 0 \text{ otherwise} \end{cases} \\
z_i &= x_i\beta + \epsilon_i \\
\epsilon_i &\sim \mathcal{N}(0,1) \\
\beta &\sim \pi(\beta)
\end{aligned}
\tag{5.6.1}
$$

where y_i is now a deterministic conditional on the sign of the stochastic auxiliary z_i. Under the independence of ϵ_i, $i = 1, \ldots, n$, the marginal likelihood of $L(\beta|y)$ in model (5.6.1) is the same as (5.5.1).

The advantage of working with representation (5.5.1) is that with a suitable choice of $\pi(\beta)$, efficient sampling can be performed using the Gibbs sampler as reported in Albert and Chib [1993], hence forth $A\&C$. For the case of a normal prior imposed on β, the full conditional distribution of β is still normal.

$$
\begin{aligned}
\beta|z &\sim N(B, V) \\
B &= V(v^{-1}b + x'z) \\
V &= (v^{-1} + x'x)^{-1}
\end{aligned}
\tag{5.6.2}
$$

where $x = (x_1', x_2', \ldots, x_n')'$. The full conditional of each element is then

truncated normal,

$$z_i|\beta, x_i, y_i \propto \begin{cases} \mathcal{N}(x_i\beta, 1)I(z_i > 0) & \text{if } y_i = 1 \\ \mathcal{N}(x_i\beta, 1)I(z_i \leq 0) & \text{otherwise} \end{cases} \tag{5.6.3}$$

which is straightforward to sample from (see for example Robert [1995])

This auxiliary variable approach offers a convenient framework for Markov Chain simulation by iteratively sampling from conditional densities in (5.6.2) and (5.6.3). There exists however a potential correlation between β and z, clearly shown in model (5.6.1) . Under the standard $A\&C$ iterative updating, this correlation is likely to cause slow mixing in the chain

To remedy this, $H\&H$ suggested a simple approach to reduce the auto-correlation and improve the mixing in the Markov Chain. They proposed to update β and z jointly, making use of the factorisation

$$\pi(\beta, z|y) = \pi(z|y)\pi(\beta|z) \tag{5.6.4}$$

where the distribution of z is unchanged from the above in (5.6.2) but now z is updated from its marginal distribution having integrated over β. It is now assumed that the prior of β is a mean with zero normal density, $N(0, v)$. From standard matrix algebra, we than obtain

$$\pi(z|y) \propto N(0, I_n + xvx')\mathbb{1}(y, z) \tag{5.6.5}$$

where I_n denotes the nxn identity matrix and $\mathbb{1}(y, z)$ is the indicator function which truncates the multivariate normal distribution of z to the appropriate region. It is known to be difficult to sample direct from the multivariate truncated normal. It is however straight forward to Gibbs sample the distribution

$$z_i|z_{-i}, y_i \propto \begin{cases} \mathcal{N}(m_i, v_i)I(z_i > 0) & \text{if } y_i = 1 \\ \mathcal{N}(m_i, v_i)I(z_i \leq 0) & \text{otherwise} \end{cases} \tag{5.6.6}$$

where z_i denotes the auxiliary variable z with the ith variable removed. The means m_i and variances v_i, $i = 1, \ldots, n$ are obtained using the leave-one-out marginal predictive densities. Using for example Henderson and Searle [1981], the parameters can be calculated efficiently as

$$\begin{aligned} m_i &= x_i B - w_i(z_i - x_i B) \\ v_i &= 1 + w_i \\ w_i &= h_i/(1 - h_i) \end{aligned} \tag{5.6.7}$$

where z_i is the current value of z_i, B is taken from (5.6.2) and h_i is the ith diagonal element of the Bayesian hat matrix $h_i = (H)_{ii}$, $H = xVx'$ with V defined in (5.6.2)

Following an update to each z_i the posterior mean B is recalculated using the relationship

$$B = B^{old} + S_i(z_i - z_i^{old}) \tag{5.6.8}$$

where B_{old} and z_{old} denotes the values of B and z_i prior to the update of z_i and S_i denotes the ith column of $S = Vx'$

Note that the variance v_i in (5.6.7) is always greater than one, the variance of the conventional iterative sampler. During the simulation, the calculation of S ,w_i and v_i has to be performed only once prior to the MCMC loop.

This algorithm carries little increase in computational burden over the conventional burden. And the use of joint updating improves mixing and sampling efficiency in the Markov Chain as demonstrated with $H\&H$

5.7 Logistic regression with auxiliary variables

Consider the model in (5.6.1). Suppose we now take $\epsilon_i \sim \pi(\epsilon_i)$ to be the standard logistic distribution, we obtain the logistic regression model but the conditional conjugacy of updating β is lost. However, if we introduce a further set of variables with the additional variables λ_i, $i = 1, \ldots, n$ and take note of the additional representation

$$
\begin{aligned}
y_i &= \begin{cases} 1 \text{ if } z_i > 0 \\ 0 \text{ otherwise} \end{cases} \\
z_i &= x_i\beta + \epsilon_i \\
\epsilon_i &\sim \mathcal{N}(0, \lambda_i) \\
\lambda_i &= (2\psi_i)^2 \\
\psi_i &\sim KS \\
\beta &\sim \pi(\beta)
\end{aligned}
\tag{5.7.1}
$$

where ψ_i, $i = 1 \ldots, n$ are independent normal variables following the Kolmogorov-Smirnov (KS) distribution [Devroye, 1986]. In this instance ϵ_i has a scale mixture of normal from with a marginal logistic distribution Andrews and Mallows [1974] so that the marginal likelihood $L(\beta|y)$ for models (5.7.1) and (5.6.1) with logit link are equivalent.

As before, a normal prior is imposed on β, $\pi(\beta) = N(b, v)$ the full conditional distribution of β given z and λ is still normal,

$$
\begin{aligned}
\beta|z &\sim N(B, V) \\
B &= V(v^{-1}b + x'Wz) \\
V &= V(v^{-1} + x'Wx)^{-1} \\
W &= diag(\lambda^{-1}, \ldots, \lambda_n^{-1})
\end{aligned}
\tag{5.7.2}
$$

and the full conditional of z_i is truncated normal, but with individual variances λ_i

$$
z_i|\beta, x_i, y_i, \lambda_i, \propto \begin{cases} \mathcal{N}(x_i\beta, \lambda_i)I(z_i > 0) \text{ if } y_i = 1 \\ \mathcal{N}(x_i\beta, \lambda_i)I(z_i \leq 0) \text{ otherwise} \end{cases} \tag{5.7.3}
$$

Finally the conditional distribution of π does not have a standard form. However it is simple to generate from using rejection sampling as outlined in Appendix A

The above framework allows for automatic sampling from the Bayesian logistic regression model using iterative updates, say $(\beta|z, \lambda)$ followed by $(z|\beta, \lambda)$ and then $(\lambda|z, \beta)$. This sampling will be slower then in the probit case as not only λ has to be sampled but but also the posterior variance matrix V in (5.7.2) for each change update of λ.

As before joint updating can be used to improve the performance of MCMC sampler. There are two possible options. One is the following the procedure in (5.6) and update z, β jointly given λ,

$$\pi(z, \beta|y, \lambda) = \pi(z|y, \lambda)\pi(\beta|z, \lambda) \tag{5.7.4}$$

followed by an update to $\lambda|z, \lambda$. The pseudo-code for this method is given in Appendix A. On the other hand, $\{z, \lambda\}$ can be updated jointly given β

In the latter the marginal densities for the $z_i's$ are independent truncated logistic distributions

$$z_i|\beta, x_i, y_i, \propto \begin{cases} \text{Logistic}(x_i\beta, 1)\mathbb{1}(z_i > 0) \text{ if } y_i = 1 \\ \text{Logistic}(x_i, \beta, 1)\mathbb{1}(z_i \leq 0) \text{ otherwise} \end{cases} \tag{5.7.5}$$

where the Logistic(a, b) denotes the density of the logistic distribution with mean a and scale parameter b [Devroye, 1986] The advantage of this latter approach is that sampling from the truncated logistic distribution can be done efficiently by the inversion method because both the distribution function and it inverse have simple analytics forms The pseudo code is given in Appendix A

5.8 Polychotomous Logistic Regression

All these previous methods can be extended to the polychotomous case by considering the conditional likelihood of a set of coefficients say β_j, having fixed the other coefficients in the model $\beta_{-j} = \{\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots, \beta_Q\}$ in

the model. In this instance

$$
\begin{aligned}
L(\beta_j|y,\beta_{-j}) &\propto \prod_{i=1}^{n}\prod_{k=1}^{Q}[\theta_{ik}]^{I(y_i=k),} \\
&\propto \prod_{i=1}^{n}[\omega_i\eta_{ij}]^{I(y_i=j),}[\omega_i(1-\eta_{ij})]^{I(y_i\neq j)}, \\
&\propto \prod_{i=1}^{n}[\eta_{ij}]^{I(y_i=j),}[\omega_i(1-\eta_{ij})]^{I(y_i\neq j)},
\end{aligned}
$$

where $\mathbb{1}(\cdot)$ is the logical indicator function and ω_i is a weight function independent of β_j

$$\eta_{ij} = \frac{\exp(x_i\beta_j - C_{ij})}{1+\exp(x_i\beta_j - C_{ij})} \tag{5.8.1}$$

$$C_{ij} = log\sum_{k\neq j}\exp(x_i\beta_k) \tag{5.8.2}$$

The conditional likelihood $L(\beta_j|y,\beta_{-j})$ has the form of a logistic regression on class indicator $\mathbb{1}(y_i = j)$. This allows the use of the logistic technique highlighted in (5.7) embedded within a Gibbs step looping over $Q-1$ classes. Appendix A lists the pseudo code, generalising the logistic scheme

5.9 Assessing Convergence

The method we have elected to assess convergence in this paper in the method of Gelman and Rubin [1992a,b]. Their method works as follows: suppose we have parallel chains starting from different points, then we can compare the variability within and between runs and when the between-run variability has reduced to what can be predicted from within-run variability, all runs should be close to equilibrium. After convergence, all chains should exhibit same qualitative and quantitative behaviour. The algorithm starts by initializing the chains at points that are overdispersed with respect to the posterior distribution. The number of chains need not be too large to avoid computational waste and are usually given in single digit numbers. Ripley [1987] suggested 3.

Gewke [1992] elaborated on the idea on how one can assess for convergence of the chain by testing whether dispersion within the chains is larger than dispersion between chains. Suppose we have a real function $\psi = t(\theta)$ and we have m trajectories $\psi_i^1, \psi_i^2, \ldots, \psi_i^n,\ i = 1, 2, \ldots, m$. The variances within and between chains W and B are given by

$$B = \frac{n}{m-1} \sum_{i=1}^{m} (\bar{\psi}_i - \bar{\psi})^2)$$

$$W = \frac{1}{n(m-1)} \sum_{i=1}^{m} \sum_{j=1}^{n} (\psi_i^j - \bar{\psi}_i)^2$$

where $\bar{\psi}_i$ is the average of the chain $i, i = 1, 2, \ldots, m$ and $\bar{\psi}$ is the average

of these averages. If convergence has been attained, all these mn values are drawn from the posterior and σ^2_ψ the variance of ψ can be consistently estimated by W, B and the weighted average $\hat{\sigma}^2_\psi = (1 - 1/n)W + (1/n)B$

If convergence has yet to be attained, the trajectories of the chain will still be highly influenced by the initial values and due to their overdispersion, will cause $\hat{\sigma}^2$ to overestimate σ^2. W, on the other hand , will underestimate σ^2 because the chains will not have adequately transversed the complete state space. Hence an indicator of convergence can be formed by using the estimator of potential scale reduction $\hat{R} = \sqrt{\hat{\sigma}^2_\psi / W}$, which will always be larger than 1. As $n \to \infty$ by the ergodic theorem both estimators converge to σ^2 and R to 1 . Gelman et al. [1996] suggested accepting convergence when R is below 1.2.

Gelman's and Rubin's method of convergence was designed for a univariate measure but our feature vectors are multivariate. While there is a multivariate version of the Gelman and Rubin method [Brooks and Gelman, 1998], it still essentially relies on the concept of transforming a multivariate measure to a univariate measure. We have chosen to use logarithmic scoring [Good, 1983] which sums the negative log probability of the event as a univariate measure. In the R package coda, the function gelman.diag can be used to perform the Gelman and Rubin diagnostic.

5.10 Comparison with other machine learning models

Table 5.1: Comparing cross-validated results with other models

Model/Data Set	**Crabs**	**Music**	**Crystals**	**Glass**	**Wave**	**Zip**
Neural Nets	30.8 (5.8)	23.6 (5.9)	20.4 (3.4)	29.1 (0.8)	13.0 (3.7)	6.7(4.0)
Random Forest	36.0 (7.9)	25.3 (5.6)	20.7 (4.2)	21.3 (6.4)	13.9 (2.6)	4.5(3.6)
C5.0	35.3 (7.9)	25.8 (5.6)	22.1(5.5)	23.0 (7.2)	15.6 (3.0)	4.0 (3.4)
SVM-Poly	34.5 (7.1)	22.3 (6.3)	20.2 (4.5)	31.5 (6.7)	12.8 (3.0)	2.0 (2.7)
SVM-Rad	31.0 (8.4)	25.9 (6.5)	21.0 (3.7)	34.4 (7.7)	12.6 (3.2)	3.3 (2.8)
MDA	39.3 (1.5)	29.1 (8.3)	23.0 (4.7)	33.5 (7.8)	13.2 (3.1)	6.5 (3.2)
ME-Freq	28.7 (1.5)	18.9 (1.3)	18.9 (7.2)	25.2 (2.2)	12.8 (3.1)	3.5 (3.2)
ME-Baye	27.5 (1.0)	16.5 (0.7)	16.2 (2.5)	22.9 (1.1)	12.2 (1.2)	2.9 (0.7)

Table 5.2: Comparison with other models using a statistical test

Model/Data Set	**Crabs**	**Music**	**Crystals**	**Glass**	**Wave**	**Zip**
Neural Nets	+	+	+	+	+	+
Random Forest	+	+	+	-	+	+
C5.0	+	+	+	0	+	+
SVM-Poly	+	+	+	+	+	-
SVM-Rad	+	+	+	+	0	+
MDA	+	+	+	+	+	+
ME-Freq	+	+	+	+	+	+
ME-Bayes	+	+	+	+	+	+

Again we perform the 5×2cv test for our model against the alternative methods as well the frequentist approach for mixture of experts and the results are given below. The Bayesian approach outperformed frequentist approach on all the datasets. For the cases where the frequentist approach failed to outperform the alternative, the Bayesian approach has enabled us to achieve parity with the C5.0 on the glass data set, still underperforming random forest, outperforming both the SVMS on the waveform dataset, outperforming SVM (radial kernel) on Zip code data set but still underperforming support vector machine with the RBF kernel.

Chapter 6

Blending Generative and Discriminative Methods

6.1 Introduction

Discriminative methods provide excellent predictive performance and are widely used in many applications. Recently there has been a growing interest in a complementary approach based on generative models. One of the motivations is that in complex problems such as object recognition, where there exists huge variability in the range of possible input vectors, it might be difficult to provide enough labeled training examples. Hence semi-supervised learning, in which labelled training examples are supplemented with a much larger quantity of unlabeled examples, are growing in popularity. As highlighted earlier, a discriminative approach cannot utilise unlabeled data and

hence a generative approach has to considered

The complementary strengths of generative and discriminative models have let a number of researchers to develop methods which combine their strengths. There has been a growing interest in 'discriminative training' of generative models with the aim of improving classification accuracy [Bouchard and Bill Triggs, 2004, Holub and P. Perona, 2005, Yakhnenko et al., 2005]. This approach has been widely used in speech recognition with great success where generative hidden Markov Models are trained by optimizing the predictive conditional distribution. As will be shown later, this form of training can lead to improved performance by compensating for model mis-specification i.e. the difference between the distribution specified by the model and the true distribution of the process which generates the data. However as pointed out previously, discriminative approach cannot make use of unlabeled data. It has been shown that Ng and Jordan [2002] that logistic regression works much better than its generative counterpart, but only when there are a large number of training points, which necessitates the use of unlabeled data.

Recently several researchers [Bouchard and Bill Triggs, 2004, Holub and P. Perona, 2005, Raina et al., 2003] have proposed hybirds of the generative and discriminative approaches in which a model is trained by optimizing a convex combination of the generative and discriminative log likelihood functions. While this is highly heuristic, it was discovered that the best predictive performance was obtained for intermediate regions between the generative and discriminative limits.

Lasserre and Bishop [2007] developed a novel viewpoint [Minka, 2005, Bishop, 2006] which proposed, for a given model, there is a unique likelihood function and hence there is only one correct way to train it. The "discriminatively" training of a generative model is interpreted as the standard training of a different model, corresponding to the choice of a distribution. This removes the adhoc choice for the training criterion, so all models are trained according to the principles of statistical inference. In addition, by the introduction a constraint between the parameters of the model, the original generative model can be recovered.

This perspective, in addition to giving a novel interpretation of "discriminative' training", opens the way to principled lending of blending generative and discriminative via the introduction of priors having a soft constraint among parameters. The strength of this constraint governs the balance generative and discriminative.

6.2 A New View of Discriminative Training

A generative model can be defined by specifying the joint distribution $f(x, c|\theta)$ of the input vector x and the class label c, conditioned on a set of parameters θ. This is typically done by defining a prior probability of the classes $f(c|\pi)$ along with a class conditional density for each class $f(x|c, \lambda)$, so that

$$f(x, c|\theta) = f(c|\pi)f(x|c, \lambda) \tag{6.2.1}$$

where $\theta = \{\pi, \lambda\}$. Since it is assumed that the data points are independent, the joint distribution is given by

$$L_G(\theta) = f(X, C, \theta) = f(\theta) \prod_{i=1}^{N} f(x_n, c_n|\theta) \tag{6.2.2}$$

This can be maximised to determine the most probable (MAP) value of θ. Again since $f(X, C, \theta) = f(X, C, \theta) f(X, C)$, this is equivalent to maximising the posterior distribution $f(\theta|X, C)$

To improve the predictive performance of generative models, it has been proposed to use discriminative training [Yakhnenko et al., 2005] which involves maximising

$$L_D(\theta) = f(C, \theta|X) = f(\theta) \prod_{i=1}^{N} f(c_n|x_n, \theta) \tag{6.2.3}$$

in which the input vectors are conditioned on instead of modeling their distributions. Here they have used

$$f(c|x, \theta) = \frac{f(x, c|\theta)}{\sum_{c'} f(c_n|x_n, \theta)} \tag{6.2.4}$$

Note that (6.2.3) is not the joint distribution of the original model defined by (6.2.2) and hence does not correspond to the MAP for this model. Hence the terminology 'discriminative training' is misleading since for a given model there is only one way to change one correct way to train it. Hence it is not the training model which has changed, but the model itself

This idea of discriminative training has been extended further by Yakhnenko et al. [2005] by maximising a function given a convex combination of (6.2.3) and (6.2.2) of the form

$$\alpha \log L_D(\theta) + (1 - \alpha) log L_G(\theta) \tag{6.2.5}$$

where $0 \leq \alpha \leq= 1$ so as to interpolate between the generative ($\alpha = 0$) and ($\alpha = 1$) approaches. Unfortunately this criterion is not derived by maximising the distribution of a well-defined model

Following Minka [2005], Lasserre and Bishop [2007] proposed an alternative view of discriminative training, which results in a elegant framework for blending generative and discriminative approaches. Consider a model which obtains a additional independent set of parameters $\tilde{\theta} = \{ \tilde{\pi}, \tilde{\lambda}\}$ in additional parameter $\theta = \pi, \lambda$ which the likelihood function is given by

$$q(x, c|\theta, \tilde{\theta}) = f(c|x, \theta) f(x| \tilde{\theta}) \tag{6.2.6}$$

where

$$f(x|\theta) = \sum_{c'} f(c|x, \theta) \tag{6.2.7}$$

Here $f(c|x, \theta)$ is defined by (6.2.7) while $f(x, c|\tilde{\theta})$ has independent parameters θ.

The model is completed by defining a prior $f(\theta, \tilde{\theta})$ over the model param-

eters giving a joint distribution of the form

$$q(X, C, \theta, \tilde{\theta}) = f(\theta, \tilde{\theta}) \prod_{n=1}^{N} f(c_n|x_n, \theta) f(x_n, \tilde{\theta}) \tag{6.2.8}$$

Now Suppose we consider a special case in which the prior factorizes, so that

$$f(\theta, \tilde{\theta}) = f(\theta,\ \theta)) \tag{6.2.9}$$

The optimal values of parameters θ and $\tilde{\theta}$ are determined in the usual way by maximizing (6.2.6), which now takes the form

$$[f(\theta) \prod_{n=1}^{N} f(c_n|x_n\theta) f(\tilde{\theta})][\prod_{n=1}^{N} f(c_n|x_n\tilde{\theta})] \tag{6.2.10}$$

The resulting value of θ will be identical to that found by maximising (6.2.7) since it is the same function which is being maximised. Since it is θ and $\tilde{\theta}$ which determines the predictive distribution $f(c|x, \theta)$ we that this model is equivalent in its predictions to "the discriminative trained" generative model. This results in a consistent view of training in which the joint distribution is always maximised and the distinction between generative and discriminative training lies in the choice of the model.

Now suppose we consider a prior which enforces the equality between the two sets of parameters

$$f(\theta, \tilde{\theta}) = f(\theta)\delta(\theta - \tilde{\theta}). \tag{6.2.11}$$

Then we can set $\tilde{\theta} = \theta$ in which the original generative model $f(x, c|\theta)$ is recovered. Hence we have a single class of distributions which corresponds to independence in the prior and the generative model corresponds to an equality constraint in the prior.

6.3 Blending Generative and Discriminative

It is now possible to blend between generative and discriminative extremes by considering priors which impose a soft constraint between $\tilde{\theta}$ and $\tilde{\theta}$. What is the rationale of doing this?

Firstly it is noted that the reason why "discriminative training" might give better results than direct use of generative model is that (6.2.8) is more flexible than (6.2.2) since it relaxes the implicit constraint $\tilde{\theta} = \theta$. If the generative model is a perfect representation of reality, then increasing the flexibility of the model would lead to poorer results. Hence any improvement from the discriminative approach must can only be from the result of mismatch between the model and the true-distribution of the data. Benefit of 'discriminative training' results from compensating for model mis-specification.

Conversely, the strength of the generative approach it can utilize unlabeled data to supplement the labeled training set. Assuming now we have

data set comprising of a set of inputs X_L for which we have corresponding labels C_L, together with a set of inputs X_U for which we have no labels. For the correctly trained generative model, the function which is maximized is given by

$$f(\theta) = \prod_{n \in L} f(x_n, c_n|\theta) \prod_{m \in U} f(x_m, |\theta) \tag{6.3.1}$$

where $f(x|\theta)$ is defined by $f(x|\theta) = \sum_{c'} f(x, c|\theta)$

We see that the unlabeled data influences the choice of θ and hence the model predictions. In contrast, for the "discriminatively trained" generative model the function which is again the product of the prior and the likelihood and hence takes the form

$$f(\theta) = \prod_{n \in L} f(x_n, |x_c\theta) \tag{6.3.2}$$

and it is observed that unlabeled data plays no role. Hence in order to make use of unlabeled data a discriminative approach cannot be used.

To see how combination of labeled and unlabeled data can be exploited from the perspective approach defined by (6.2.8), for which the joint distribution now becomes

$$q(X_L, C_L, X_U, \theta, \tilde{\theta}) = f(\theta, \tilde{\theta}) \left[\prod_{n \in L} f(c_n|x_n, \theta) f(x_n|\tilde{\theta}) \right] \left[\prod_{m \in U} f(x_m|\tilde{\theta}) \right] \tag{6.3.3}$$

We can see that the unlabeled data now influence the parameter $\tilde{\theta}$m which in turn influences θ which in turn influence θ via the soft constrain imposed by the prior.

Generally, if the model is not a perfect representation of reality, and unlabeled data is available , then we would expect the optimal balance to lie neither at the purely generative extreme or at the purely discriminative extremes

A simple example of a prior which interpolates smooth between the generative and discriminative limits, consider the class of priors of the form

$$f(\theta, \tilde{\theta}) \propto f(\theta) f(\tilde{\theta}) \frac{1}{\sigma} \exp\left\{ -\frac{1}{2\sigma^2} ||\theta - \tilde{\sigma}||^2 \right\} \tag{6.3.4}$$

if desired, σ can be related to α by defining a map from $(0, 1)$ to $(0, \infty)$ for example using

$$\sigma(\alpha) = \left(\frac{\alpha}{1 - \alpha} \right)^2 \tag{6.3.5}$$

For $\alpha \to 0$ and $\sigma \to 0$ a constraint of the form (6.2.11) is obtained which corresponds to the generative model. Conversely for $\alpha \to 1$ and $\sigma \to \infty$, we obtain an independence prior of the form (6.2.9)

Chapter 7

Conclusion

In this thesis, we have used mixture of experts in conjunction with logistic regression to classify partially labelled data. The most commonly used parametric method of classifying such data is by using a mixture of normals. By relying on the assumption of Gaussian densities (more information compared to logistic discrimination), results in an efficient (less variance) discrimination procedure if the modelling assumptions are satisfied. However in practice, these assumptions very rarely holds and very often some of the features are qualitative variables, and hence it is generally of the view that logistic discrimination is a safer and more robust bet as it relies on fewer assumptions.

Mixture of expert models have been around since early 1990s but did not gain widespread usage. There seem to be several reasons, the main one being that mixture of experts are much harder to fit compared to other type

of statistical models and due to this difficulty there wasn't a general purpose implementation of mixture of experts models publicly available, a gap which since has been filled by [Evers, 2007].

Despite these difficulties, mixture of experts have a lot of attractive properties. They permit a meaningful interpretation of the model parameters while still being able to model complex relationships. Most other complex methods, like neural networks and support vector-machines, are black boxes i.e. they do not allow for a meaningful interpretation of their results. On the flip side, many simple statistical models which can be interpreted easily are usually too restrictive.

Evers [2007] covered the frequentist fitting of HME. One major limitation of the maximum likelihood approach is the tendency for overfitting. This is particularly likely to be the case in mixture of experts as a huge number of parameters are involved in defining the gating and expert networks. Any one of the mixture components "collapsing" onto a single data point will result in plenty of singularities arising in the likelihood function. In addition the maximum likelihood framework provides no direct mechanism for determining either the topology the HME tree or the number of nodes since optimization of the likelihood function will always favour more complex models. These problems can be resolved by adopting a Bayesian approach which is what we have done in this thesis and empirical tests on different data sets showed that the Bayesian approach outperforms maximum likelihood estimation.

7.1 Directions for Future Research

While we have shown that mixture of experts outperformed traditional neural networks, there have been a lot of advances in neural networks in recently years. Deep learning neutral works are now very popular and have shown great results in image recognition, outperforming all known classifiers in the Zip Code data set.

Despite tremendous advances in MCMC algorithms and computing power, we still had to rely on the use of a conjugate prior as the MCMC chain still mixes too slowly when we couldn't simulate from a exact distribution and when abundant data is not available. Neal [1996] developed a Hybird Monte Carlo (HMC) Scheme for mixture of experts but HMC methods has largely been ignored by the statistical community. In recent years, there has been a revival of interest in HMC algorithms [Ishwaran, 1999, Girolami and Calderhead, 2011] and applications to logistic regression have demonstrated promising results. Without being constrained to the use of a conjugate prior, other possible priors can be experimented with to see if we can get better results.

Appendix A

Procedure for sampling the Bayesian polychotomous model

%% Let $Y[i][j]$ denote the category indicator variable $Y[i][j] = 1$ if the ith observation is of the class j, $j \in \{1, \cdots, Q\}, Y[i][j] = 0$ otherwise.

%% Initialise mixing weights, $\Lambda[,,q]$ for each category $(n \times n)$ identity matrix

FOR $q = 1$ *to* $Q - 1$

$\Lambda[,,q] \leftarrow I_n$

%% draw Z from truncated logistic

$Z[,q] \sim Lo(0,1)\mathbb{1}(Y[,q]), Z[,q])$

END

For i = 1 to number of MCMC iterations

FOR $q = 1$ *to* $Q - 1$

$V \leftarrow (X^T \Lambda[,,q]^{-1} X + v^{-1})^{-1}$

%% note that Λ^{-1} is a diagonal matrix and hence simple to invert

$L \rightarrow Chol(V)$

%% so L stores the lower triangular Cholesky factorisation of V

$B \rightarrow V X^T \Lambda[,,q]^{-1} Z[,q]$

$T \sim \mathcal{N}(0, I_p)$

$\beta[,q,i] \rightarrow B + LT$

%% Now update $\{Z, \Lambda\}$

FOR j=1 to number of observations

$m \rightarrow X[j,]\beta, q, i]))$

$C \rightarrow \ sum(exp(X[j,]\beta, -q, i]))$

%% Hence C records the sum of $Q-2$ terms, $\exp(X[j,], \beta[,t,i])$,

%% for $t \in \{1, \cdots q-1, q+1, \cdots, Q-1\}$,

Now draw $Z[j,q]$ from truncated logistic

%% now draw new value of mixing variance

$R \leftarrow Z([j,q] - m)\ R\lambda[j,j,q] \sim pi(\lambda | R^2)$

%% See next section

END

END

End MCMC iterations; RETURN β

Now we describe how to sample from the full conditional distribution of the auxiliary weights $\Lambda = \{\lambda_1, \cdots, \lambda_n\}$ in the logistic regression model of

5.7. This conditional distribution however does not have a standard form but sampling from the density can be achieved efficiently using rejection sampling.

As a rejection sampling density it is suggested to use the Generalized Inverse Gaussian distribution $GIG(\lambda, \psi, \chi)$ Using the parameterisation of Devroye [1986] we set $\lambda = 0.5$, $b\psi = 1$ and $\chi = (z_i - x_i\beta)^2 = R^2$. When sampling from the GIG we make use of the equality $GIG(0.5, 1, r^2) = r/IG(1, r)$, where IG denotes the inverse Gaussian. It is easier to sample from than the GIG as it can be done from an inversion algorithm [Devroye, 1986]

Let $g(\lambda)$ denote $GIG(0.5, 1, r^2)$ rejection sampling density $r^2 = (z_i - x_i\beta)^2$. Following a draw from $g(\cdot)$ the sample is accepted with probability $\alpha(\cdot)$

$$\alpha(\lambda) = \frac{l(\lambda)\pi(\lambda)}{Mg(\lambda)}, \tag{A.0.1}$$

where $M \geq sup_\lambda \frac{l(\lambda)\pi(\lambda)}{Mg(\lambda)}$, $l(\lambda)$ denotes the likelihood, $l(\lambda) \propto \lambda^{-1/2} \exp(-0.5r^2/\lambda)$ and $\pi(\lambda)$ is the prior

$$\pi(\lambda) = \frac{1}{4}\lambda^{-1/2} KS(\frac{1}{2}\lambda^{\frac{1}{2}}) \tag{A.0.2}$$

where $KS(\cdot)$ denote the Kolmogorov-Smirnov density. The prior A.0.2 follows from the transformation of the random variables $\lambda_i = (2\phi_i)^2$ in 5.7.1

We can set $M = 1$ and cancelling terms leaves the acceptance probability A.0.1 as

$$\alpha(\lambda) = \exp(0.5\lambda)\pi(\lambda) \tag{A.0.3}$$

Evaluating $\alpha(\lambda)$ is problematic as the KS density is only known up to an infinite series. There is however, an alternating series representation given in Devryone (1986) that allows for an efficient set of squeezing functions to be adopted for the rejection sampling algorithm. Following Devroye (1986), λ can be partition into two regions and a monotone alternative series can be constructed.The breakpoint for his mixture method can be anywhere in the interval $[4/3, \pi^2]$. We have used the value $4/3$ as the rightmost interval is faster to evaluate.

The pseudo code is detailed below. The method is least efficient as $r_i^2 \to 0$, though it is possible to observe an acceptance rate of around 0.25 for r_i^2 as small as 10^{-10}. For $r_i^2 = 1$ the acceptance is around 0.5 rising to nearly one for $r_i^2 \geq 1$

Procedure to sample $\lambda \sim \pi\lambda|r^2)$

REPEAT

%% Note, $r^2 = (z_i - x_i\beta)^2$

%% To begin we must draw a sample from the rejection sampling density

$Y \sim \mathcal{N}(0, 1)$

$Y \leftarrow Y^2\ Y1 + (Y - \sqrt{Y(4r + Y)})/2r$

$U \sim U[0, 1]$

IF $U \leq 1/(1+Y)$THEN $\lambda \leftarrow r/Y$

ELSE $\lambda \leftarrow rY$

%% Now, $\lambda \sim GIG(0.5, 1, r^2)$

$\mathcal{U} \sim U[0,1]$

IF $\lambda \geq 4/3$

OK $\rightarrow$ rightmost-intervale(U, λ)

ELSE

OK $\rightarrow$ rightmost-intervale(U, λ)

WHILE NOT OK

The procedure above calls for two functions, rightmost-interval() and leftmost-interval(), depending on the value of proposed λ. The pseudo code of these functions follows:

OK $\leftarrow$ rightmost-interval (U, λ)

$Z \leftarrow 1$

$X \leftarrow \exp(-0.5\lambda)$

$j \leftarrow 0$

REPEAT

%% Squeezing

$j \leftarrow j + 1$

$Z \leftarrow Z - (j+1)^2 X^{(j+1)^2} - 1$

$IF Z \geq U$ THEN RETURN OK $\leftarrow 1$

$j \leftarrow j + 1$

$Z \leftarrow Z + (j+1)^2 X^{(j+1)^2} - 1$

IF $Z < U$ THEN RETURN OK $\rightarrow 0$

END

The pseudo-code for the left region is

OK $\leftarrow$ leftmost-interval (U, λ)

$H \leftarrow 0.5 \log(2) + 2.5 log(\pi) - 2.5 log(\lambda) - \frac{\pi^2}{2\lambda} + 0.5\lambda$

$lU \leftarrow log(U)$

$Z \leftarrow 1$

$X \leftarrow exp(-\pi^2/(2\lambda))$

$K \leftarrow \lambda\pi^2$

$j \leftarrow 0$

REPEAT

%% Squeezing

$j \leftarrow j + 1$

$Z \leftarrow -KX^{j^2-1}$

IF $H + \log(Z) \geq lU$ THEN RETURN OK $\rightarrow 1$

$j \leftarrow j + 1$

$Z \leftarrow -(j+1)^2 X^{(j+1)^2-1}$

IF $H + \log(Z) \leq lU$ THEN RETURN OK $\rightarrow 0$

END

Bibliography

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422): pp. 669–679, 1993.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1): 99–102, 1974. ISSN 00359246.

C. Andrieu and J. Thoms. A tutorial on adaptive mcmc. *Statistics and Computing*, 18(4):343–373, 2008.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

G. Bouchard and Bill Triggs. The trade-off between generative and discriminative classifiers. In *Proceedings in Computational Statistics, 16th Symposium of IASC*, pages 721–728. Physica-Verlag, 2004.

L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

J. Bridle. Probabilistic interpretation of feedforward classification network outputs with relationships to statistical pattern recognition. In F. Fogelman-Soulie and J. Herault, editors, *Neuro-computing: Algorithms, Architectures*, pages 227–236. New York :Springer-Verlag, 1989.

S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of the American Statistical Association*, 7(7):434–455, 1998.

W. Buntine and A. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

N. A. Campbell and R. J. Mahon. A multivariate study of variation in two species ofrock crab of genus *Leptograpsus*. *Australian Journal of Zoology*, 22:417–425, 1974.

K. Chen, L. Xu, and H. Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks*, 12(9): 1229 – 1252, 1999. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/S0893-6080(99)00043-X. URL `http://www.sciencedirect.com/science/article/pii/S089360809900043X`.

M.-H. Chen and D. K. Dey. Bayesian modeling of correlated binary responses via scale mixture of multivariate normal link functions. *Sankhya-: The Indian Journal of Statistics, Series A*, 60(3):pp. 322–343, 1998.

A. P. Dawid. Properties of diagnostic data distributions. *Biometrics*, 32(3): 647–58, 1976.

R. C. de Amorim and C. Hennig. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 324:126 – 145, 2015. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2015.06.039.

P. Dellaportas and A. F. M. Smith. Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 42(3):pp. 443–459, 1993.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 38:1–38, 1977.

L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923, oct 1998. ISSN 0899-7667.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybird Monte Carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

L. Evers. *Model Fitting and Model Selection for "Mixture of Experts" Models*. PhD thesis, University of Oxford, Oxford, United Kingdom, 2007.

P. Fengchun, R. A. Jacobs, and M. A. Tanner. Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journal of the American Statistical Association*, 91 (435):pp. 953–960, 1996.

C. Fraley. and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8): 578–588, 1998. doi: 10.1093/comjnl/41.8.578.

C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca. *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*, 2012.

J. Fritsch, M. Finke, and A. Waibel. Adaptively growing hierarchical mixtures of experts. In *In Advances in Neural Information Processing Systems 9*, pages 459–465. MIT Press, 1997.

S. Fruhwirth-Schnatter and R. Fruhwirth. Auxiliary mixture sampling with applications to logistic models. *Computational Statistics and Data Analysis*, 51(7):3509–3528, April 2007.

D. Gamerman. Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7(1):57–68, 1997. ISSN 0960-3174.

A. Gelman and D. Rubin. A single series from the gibbs sampler provides a false sense of security. *Bayesian Statistics*, 4:625–631, 1992a.

A. Gelman and D. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992b.

A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient metropolis jumping rules. In J. M. Bernado et al., editors, *Bayesian Statistics*, volume 5, page 599. OUP, 1996.

J. Gewke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In J. M. Bernado et al., editors, *Bayesian Statistics*, volume 4, pages 557–87. Oxford University Press, 1992.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41 (2):pp. 337–348, 1992.

M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

I. J. Good. *Good Thinking - The Foundations of Probability and its Applications*. University of Minnesota Press, Minneapolis, 1983.

C. Goutte. On clustering fmri time series. *NeuroImage*, 9(3):298 – 310, 1999. ISSN 1053-8119. doi: http://dx.doi.org/10.1006/nimg.1998.0391.

H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7(2):pp. 223–242, 2001.

I. Hampshire, B. John, and A. Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(7):751–769, July 1992.

T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1): 155–176, 1996. ISSN 00359246.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, 2001.

H. V. Henderson and S. R. Searle. On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1):53–60, 1981. ISSN 00361445. URL `http://www.jstor.org/stable/2029838`.

C. Holmes and K. Held. Bayesian Auxiliary Variable Models for Binary and Multinomial regression. *Bayesian Analysis*, 1:145–168, 2006.

A. D. Holub and P. Perona. A discriminative framework for modeling object class. *Computer Vision and Pattern Recognition (CVPR)*, 2005.

H. Ishwaran. Applications of hybrid monte carlo to bayesian generalized linear models: Quasicomplete separation and neural networks. *Journal of Computational and Graphical Statistics*, 8:779–799, 1999.

R. A. Jacobs and M. I. Jordan. Adaptive mixtures of local experts. *Neural Computation*, 11:79–87, 1991.

R. A. Jacobs, M. Tanner, and P. Fengchun. Bayesian inference for hierarchial mixture-of-experts with applications to regression and classification. *Statistical Methods in Medical Research*, 5:375–390, 1996.

R. A. Jacobs, P. Fengchun, and M. Tanner. A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures. *Neural Networks*, 10(2):231–241, 1997.

A. E. R. Jeffrey D. Banfield. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993. ISSN 0006341X, 15410420.

M. I. Jordan and R. A. Jacob. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. ISSN 0899-7667.

M. I. Jordan and R. A. Jacobs. Hierarchies of adaptive experts. In J. Moody and S. Hanson, editors, *Advances in Neural Information Processing Systems 4*, pages 985–993, San Mateo CA, 1992. Morgan Kaufmann.

M. I. Jordan and L. Xu. Convergence results for the {EM} approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409 – 1431, 1995. ISSN 0893-6080.

D. J. Ketchen and C. L. Shook. The application of cluster analysis in strategic management research: An analysis and critique. *Strategic Management Journal*, 17(6):441–458, 1996. ISSN 01432095, 10970266. URL http://www.jstor.org/stable/2486927.

M. Kuhn. *caret: Classification and Regression Training*, 2012. URL http://CRAN.R-project.org/package=caret. R package version 5.15-044.

C. M. B. J. Lasserre and C. M. Bishop. Generative or discriminative? getting the best of both worlds. *BAYESIAN STATISTICS*, 8:3–24, 2007.

Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Advances in neural information processing systems 2. chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-100-7. URL http://dl.acm.org/citation.cfm?id=109230.109279.

P. Lenk and W. DeSarbo. Bayesian inference for finite mixtures of generalized linear models with random effects. *Psychometrika*, 65(1):93–119, March 2000.

R. Lleti, M. C. Ortiz, L. A. Sarabia, and M. S. Snchez. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87 – 100, 2004. ISSN 0003-2670.

Papers presented at the 5th {COLLOQUIUM} {CHEMIOMETRICUM} {MEDITERRANEUM}.

P. McCullagh and J. A. Nelder. *Generalized linear models*. London: Chapman & Hall, 1989.

T. P. Minka. Discriminative models, not discriminative training. Technical Report TR-2005-144, Microsoft Research, Cambridge, 2005. URL `ftp://ftp.research.microsoft.com/pub/TR/TR-2005-144.pdf`.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., 1996. ISBN 0387947248.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

S.-K. Ng and G. J. McLachlan. Using the em algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. *IEEE Transactions on Neural Networks*, 15(3):738–749, May 2004. ISSN 1045-9227. doi: 10.1109/TNN.2004.826217.

S.-K. Ng and G. J. McLachlan. Extension of mixture-of-experts networks for binary classification of hierarchical data. *Artificial Intelligence in Medicine*, 41(1):57 – 67, 2007. ISSN 0933-3657.

S.-K. Ng, G. J. McLachlan, and A. H. Lee. An incremental em-based learning approach for on-line prediction of hospital resource utilization. *Artif. Intell. Med.*, 36(3):257–267, Mar. 2006. ISSN 0933-3657.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006. ISBN 978-0387303031.

S. J. Nowlan. *Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1991.

S. M. Obrien and D. B. Dunson. Bayesian multivariate logistic regression. *Biometrics*, 60:739–746, 2004.

J. C. Pinheiro and D. M. Bates. *Mixed-effects models in S and S-PLUS*. Springer, New York, NY [u.a.], 2000. ISBN 0387989579 9780387989570 9781441903174 1441903178.

N. G. Polson, J. Scott, and J. Windle. Bayesian inference for logistic models using polya-gammalatent variables. *ArXiv e-prints*, may 2012.

R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 545–552, 2003.

R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood, and the EM algorithm. *Siam Review*, 26:195–239, 1984.

B. D. Ripley. *Stochastic Simulation*. Wiley, New York, NY, 1987.

B. D. Ripley. Neural networks and flexible regression and discrimination. In K. V. Mardia, editor, *Statistics and Images 2*, volume 2 of *Advances in Applied Statistics*, pages 39–57, Abingdon, 1994. Carfax.

B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

C. Robert. Simulation of truncated normal variables. *Statistics and Computing*, 5(2):121–125, 1995. ISSN 1573-1375. doi: 10.1007/BF00143942. URL `http://dx.doi.org/10.1007/BF00143942`.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/0377-0427(87)90125-7. URL `http://www.sciencedirect.com/science/article/pii/0377042787901257`.

S. L. Scott. Data augmentation, frequentist estimation, and the bayesian analysis of multinomial logit models. *Statistics Papers*, 2009.

C. A. Sugar and G. M. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463):750–763, 2003. doi: 10.1198/016214503000000666.

R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953. ISSN 1860-0980. doi: 10.1007/BF02289263.

D. M. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons Ltd., 1985.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

A. Villagran and G. Huerta. Bayesian inference on mixture-of-experts for estimation of stochastic volatility. In T. B. Fomby and R. C. Hill, editors, *Advances in Econometrics*, volume 20, pages 277–296. Emerald Group Publishing Limited, 2005.

G. Wahba, G. Chong, and W. Yuedong. Soft classification, a.k.a. risk estimation, via penalized log likelihood and smoothing spline analysis of variance. In D. H. Wolpert, editor, *The Mathematics of Generalization. Santa Fe Institute Studies in the Sciences of Complexity*, page 899. Addison-Wesley, 1993.

S. R. Waterhouse. *Classification and Regression using Mixtures of Experts*. PhD thesis, Cambridge University, 1997.

S. R. Waterhouse and A. J. Robinson. Constructive algorithms for hierarchical mixtures of experts. In *NIPS'95*, pages 584–590, 1995.

C. Weihs, G. Szepannek, U. Ligges, L. Karsten, and N. Raabe. Local models in register classification by timbre. In V. Batagelj, H.-H. Bock, A. Ferligoj,

and A. Žiberna, editors, *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 315–322. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-34415-5.

M. West. Generalized linear models:outlier accommodation, scale parameters and prior distribution. *Bayesian Statistics*, 2:531–58, 1985.

J. Wilson. Automated classification of images from crystallisation experiments. In P. Perner, editor, *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, volume 4065 of *Lecture Notes in Computer Science*, pages 459–473. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-36036-0.

J. C. F. Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 03 1983.

O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *ICDM*, pages 498–505. IEEE Computer Society, 2005. ISBN 0-7695-2278-5.