A FAMILY OF VARIATIONAL ALGORITHMS FOR APPROXIMATE BAYESIAN INFERENCE OF HIGH-DIMENSIONAL DATA

by

Kehinde I. Olobatuyi

M.Sc. Statistics,
Federal University of Agriculture
Abeokuta, (2016)

Department of Statistics and Financial Mathematics
University of Milano-Bicocca

A Thesis submitted for the degree of Doctor of Philosophy of the University of Milano-Bicocca

March, 2021

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Kehinde I. Olobatuyi

Acknowledgment

With immense pleasure and deep sense of gratitude, I give all glory to God. Furthermore, I wish to express my sincere thanks to my supervisor, Dr. Robert AyKroyd, designation from the University of Leeds, Leeds, UK, without his motivation, love, and continuous encouragement, this research would not have been successfully completed. I deeply thank my tutor in person of Prof. Aldo Solari, indeed you have proven to be a tutor. Your readiness to help and support is impeccable. I sincerely appreciate your effort in bringing my PhD. program to a success. Thank you!

I am grateful to Prof. Salvatore Ingrassia and Dr. Paolo Berta from the University of Catania and University of Milano-Bicocca respectively. They motivated me to carry out the second phase of the research in the University of Milano-Bicocca and also for providing me with infrastructural facilities and many other resources needed for my research.

I would like to thank my wife Elizabeth for her love and constant encouragement and moral support along with patience and understanding. I would like to also extend my profound sense of gratitude to my parents and my siblings for all the sacrifices they made during my research and also providing me with moral support and encouragement whenever required.

Last but not the least, I wish to acknowledge the support rendered by my colleagues in several ways throughout my research work. I say "THANK YOU ALL!"

Abstract

The Bayesian framework for machine learning allows the incorporation of prior knowledge into the system in a coherent manner which avoids overfitting problems but rather seeks to approximate the exact posterior and provides a principled basis for the selection of model among alternative models. Unfortunately, the computation required in Bayesian framework is usually intractable. This thesis provides a family of Variational Bayesian framework which approximates these intractable computations with latent variables by minimizing the Kullback-Leibler divergence between the exact posterior and the approximate distribution.

Chapter 1 presents background materials on Bayesian inference, and propagation algorithms. Chapter 2 discusses the family of variational Bayesian theory. It generalizes the Expectation Maximization (EM) algorithm for learning maximum likelihood parameters. Finally, it discusses factorized approximation of Expectation propagation.

Chapter 3 - 5 derive and apply the variants of Variational Bayesian to the family of Cluster Weighted Models (CWMs). It investigates the background history of CWMs and proposes new different members into the family. First, the dimensionality of CWM is explored by introducing the t Distributed Stochastic Neighbor Embedding (tSNE) for dimensionality reduction which leads to CWMs based on tSNE for high-dimensional data. Afterwards, we propose a Multinomial CWMs for multiclass classification and Zero-inflated Poisson CWMs for zero-inflated data. This work derives and applies the EM algorithm with three different maximization step algorithms: Ordinary Least Squares (OLS), Iteratively Reweighted Least Squares (IRLS), and Stochastic Gradient Descent (SGD) to estimate the models' parameters. It finally examines the classification performance of the family of CWMs by eight different information criteria and varieties of Adjusted Rand Index (ARI).

Chapter 6 proposes a variants of Expectation Propagation: EP-MCMC, EP-ADMM algorithms to the inverse models. It demonstrates EP-MCMC and EP-ADMM on complex Bayesian models for image reconstruction and compares the performance to Markov Chain Monte Carlo (MCMC). Chapter 7 concludes with a discussion and possible future directions for optimization algorithms.

Astratto

L'approccio Bayesiano alle tecniche di machine-learning consente di integrare in un modello le informazioni a priori per evitare problemi di overfitting, cercando di approssimare la distribuzione a posteriori. Fornisce inoltre una metodologia coerente per la scelta fra diversi modelli alternativi, e richiede tipicamente uno sforzo computazionale considerevole, tale da rendere alcuni problemi intrattabili. Questa tesi propone una famiglia di metodologie di tipo Variational Bayes per approssimare la complessità computazionale dell'approccio Bayesiano tramite l'utilizzo di variabili latenti, minimizzando la distanza di Kullback-Leibler tra la distribuzione a posteriori esatta e quella approssimata. Il primo capitolo riepiloga i concetti chiave dell'inferenza bayesiana e gli algoritmi di propagazione. Il secondo capitolo introduce il metodo Variational Bayes, il quale generalizza gli algoritmi di Expectation Maximization (EM) per la stima dei parametri tramite un approccio a massima verosimiglianza. Vengono inoltre discusse le approssimazioni fattorizzate per i metodi di Expectation Propagation (EP). Nei capitoli da 3 a 5 vengono derivate e testate diverse varianti dei metodi Variational Bayes per la famiglia dei Cluster Weighted Models (CWMs) e, partendo da un breve cenno storico, vengono proposte diverse nuove classi di CWM. Inizialmente viene analizzato il problema della riduzione di dimensionalità nei CWM, introducendo una nuova classe basata su t-distributed stochastic neighbor embedding (tSNE). Nel secondo lavoro viene proposto un Multinomial CWM per la classificazione multinomiale ed un Zero-inflated Poisson CWM per dati di tipo zero-inflazionato. Vengono derivati ed applicati gli algoritmi EM per la stima dei parametri, considerando tre diverse alternative per il passo di massimizzazione: Minimi Quadrati Ordinari (OLS), Minimi Quadrati Pesati Iterati (IRLS), e Discesa Stocastica del Gradiente (SGD). Per concludere, vengono testate le performance classificative dei modelli CWM utilizzando otto criteri diversi e vari Adjusted Rand Index (ARI). Nel sesto capitolo vengono proposte due varianti del metodo di Expectation Propagation per inverse models denominate EP-MCMC e EP-ADMM, applicandole a modelli bayesiani per image-reconstruction e confrontandone le performance con i metodi MCMC. Il settimo capitolo chiude la tesi, traendo le conclusioni dei lavori svolti e riassumendo i possibili sviluppi futuri.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

Chapter 1

Introduction

1.1 Background History

In the *Pattern recognition*, *objects* are grouped into a number of categories or *classes* or *clusters*. Based on the application, these objects can be images or signal waveform or any type of measurements that needs to be classified. These objects are generally referred to as *patterns*. Pattern recognition can be dated back into the long history, but before the 1960*s*, it appeared to be the output of the theoretical research in the area of statistics. As with everything else, the evolution of computer gave rise to the demand for the practical applications of pattern recognition, which in turn set new demand for further theoretical developments. As our society experiences the evolution from the industrial to its post-industrial phase, the need for information handling and retrieval and the automation in industrial production are increasingly important. This trend has pushed pattern recognition to the high edge of today's application and research. Moreover, pattern recognition is the integral part of *machine intelligence* systems built for decision making. Pattern recognition is of utmost importance in a *machine vision* area. A machine vision system captures images via a camera and produces the description of the images captured by analyzing them. Machine vision system is most useful in the manufacturing industry for automated visual inspection. For example, manufactured objects may be allowed to pass through the inspection stage in front of the camera. Thus, images have to be analyzed online and a pattern recognition system has to classify the objects into the "defect" or "non-defect" class.

After that, an action has to be taken to modify the defected parts. In an assembly line, different objects must be located and classified in one of the classes known *a priori*.

The *text recognition* is another important area of pattern recognition with major implications in automation and information handling, e.g. Optical Character recognition (OCR) systems. An OCR system is a front-end device consisting of a light source, a scan lens, a document transport, and a detector. Light-intensity variation is translated into numbers and an image array is formed at the output of the light-sensitive detector. In a sequel, a series of image processing result into line and character segmentation. The pattern recognition software then takes over to recognize the characters, i.e., to classify each character in the correct class. Another examples of pattern recognition are online handwritting recognition systems, automatic mail-sorting.

The *computer-aided diagnosis* is another application of pattern recognition, aiming at assisting doctors in making diagnostic decisions. Although the final diagnosis is made by doctors. Computer-aided diagnosis has been applied and is a valid interest for a variety of medical data, such as X-ray, computed tomographic images, ultrasound images, electrocardiograms (ECGs), and electroencephalograms (EEGs). Computer-aided diagnosis is used for the interpretation of medical data which often depends very much on the skill of the doctors. For example, X-*ray mammography* is used as a computer-assisted diagnosis for the detection of breast cancer. Mammography, as the best method for detecting breast cancer, $10 - 30\%$ of women with breast cancer who undergo mammography have negative mammograms. Approximately, two thirds of the cases with false results were wrongly detected by the radiologist which was evident retrospectively. This however, may be due to poor image quality, eye fatigue, subtle nature of the findings. The percentage of correct classification improves at the second look by another radiologist. Thus, pattern recognition system can be developed with a sole goal of assisting the radiologists with a second reading.

Increasing confidence in the diagnosis based on mammograms would in turn reduce the number of misclassifications due to human errors.

The *speech recognition* is another area in which a great deal of research and development effort has been channeled. Speech is the natural means by which humans communicate and pass across information. Thus, the goal of building intelligence machines that recognize *spoken words* has been a long-desired one among the scientists. The potential applications of this are for example, to improve the efficiency in a manufacturing society, to remotely control the machines in hazardous environments, and to help handicapped people control machines by talking to them. A major success is to impute data into the computer by speech. Entering information by spoken words to computer is twice as fast as entry by a skilled typist. Furthermore, this can enhance our ability to communicate with deaf and dumb people.

Pattern recognition can also be applicable in the areas of *Data mining and knowledge discovery* in databases. Data mining is of wide interest in a vast range of applications such as medicine and biology, market and financial analysis, business management, science exploration, image and musical retrieval. Its fame stems from the desire that there is ever increasing demand for retrieving information and transforming it into knowledge. Moreover, this retrieved information exists in huge volume in various forms including, text, images, audio, and videos stored in different places all over the world. Traditionally, description-based model where information retrieval was based on word matching and keyword descriptions. However, one limitation about this model is low time management and high time consumption because the searching presupposes that a manual annotation of the stored information has previously been performed by human. This however, is feasible when the size of the information is not huge but impractical when the amount of the available information becomes large. Content-based retrieval systems are becoming more and more popularized where information is sought based on similarity between an objects, which is presented into the system, and objects stored in sites all over the world. For example, there are different types of content-based information retrieval (CBIR) system which are the content-based image retrieval which takes as an input a scanned image. A music content-based retrieval system takes as an input an extract from the music piece. Both systems return similar object as their inputs. The foregoing are some examples

from a much larger number of possible applications. Typically, we refer to fingerprint identification, signature authentication, text retrieval, and face recognition as all forms of pattern recognition.

1.2 Features, Feature Vectors, and Classifiers

First of, let us "mimic" a medical image classification task. Figure (1.1) shows two images, each having a distinct regions representing two classes. We see that the region of the left image from a benign class, and that of right from a malignant class. The first step is to identify the measurable quantities that make these two classes unique from each other. The preceding artificial *classification* task has outlined the rational behind a large class of pattern recognition problems. The measurements used for the classification are known as *features*. More generally, $x_l, l = 1, ..., d$ is d features that form the *feature vector*

$$\boldsymbol{x} = [x_1, x_2..., x_d]^T,$$

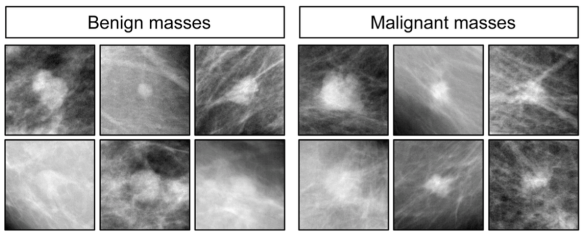


Figure 1.1: The Example of an image of cancer tissue corresponding to two classes: Benign class (Left) and Malignant class (Right)

where T denotes transposition. Each of the feature vectors identifies uniquely a single pattern (object). Throughout this thesis, features and feature vectors will be treated as *random variables* and *vectors*, respectively. This is natural, as the there is a random variation between the measurements resulting from different patterns. The variation in the measurements is due partly to the noise of the measuring device and partly to

the unique characteristics of the each pattern. For example, in X-ray imaging, wide variations are caused by the differences in physiology among individuals.

Largely, the role of the *classifiers* is to divide the feature space into regions that correspond to either class A or class B. For a *binary* categories, if an unknown pattern falls in the class A region, it is classified as class A otherwise, as class B. However, this does not mean the decision is correctly classified. If it is incorrect, a *misclassification* has occurred. The straight line across any region is called a *decision line* which explore that we knew the labels (class A or B) for each point.

1.3 Supervised, Unsupervised and Semi-supervised Learning

In *Supervised learning*, the training data is assumed to be known *a priori* where also the information about the class label is available. However, this in general might be too expensive and there is another type of pattern recognition tasks for which training data of known class labels are not available. In this type of task, we only have access to the set of feature vectors $\boldsymbol{x}$ and the goal is to discover the underlying similarities or cluster of similar vectors together. This is known as the *unsupervised pattern recognition*. Some applications of unsupervised learning are remote sensing, image segmentation, image and speech coding. A *clustering* algorithm is typically employed to reveal the inherent grouping of the feature vectors in the $\mathcal{D}$-dimensional feature space. Points that share common or similar characteristics are clustered together. Once this is done, the type of each cluster can be identified by associating a sample of points in each group with available data.

Semi-supervised pattern recognition shares the same attribute with the supervised learning discussed above. However, in semi-supervised, the analyst has a partial information about the class labels that is, a set of patterns of unknown class origin, in addition to the training patterns with known class. The former is generally referred to as *unlabeled* and the latter as *labeled* data. Semi-supervised pattern recognition can be useful when the analyst has limited information about the data, i.e., has a limited number of labeled data. In such case, recovering an additional information from the

unlabeled samples, related to the generic structure at hand, can be used to improve the design. Semi-supervised learning finds a way of clustering tasks which is constrained to assign certain points in the same cluster or different cluster. From this viewpoint, semi-supervised learning provides *a priori* information that the unsupervised learning has to respect [Jan (2005)].

1.4 Introduction to Image Reconstruction

In the original usage, medical X-rays are passed through the body, projecting images such as bones, organs, air spaces, and tumors unto a two-dimensional sheet of film. This important diagnostic tool suffer from a major limitation: superimposition in such a single X-ray projection are difficult or sometimes impractical to unravel when there exists a slim margin in the differences of the X-ray densities, as between tissue and an embedded tumor. Diagnosis now becomes more accurate through the recently developed mathematical procedures for combining X-ray projections taken at different angles around the body now make it possible to reconstruct a quantitative three-dimensional representation of the internal structure of a living human body. In addition to their contribution to medicine, the new method for reconstruction from projections has a wide range of applications such as in microscopy, X-rays, and light. In astronomical application, two-dimensional images of galaxies can be reconstructed from radio and X-ray signals by mathematically identical methods. By reconstructing only a thin (two-dimensional) slice of an image at a time, three-dimensional reconstruction can be greatly simplified. A two-dimensional cross-section can be regarded as a *picture*. In this thesis, we shall be mostly concerned with the reconstruction of two-dimensional picture from its one-dimensional projection. We first discuss how the object and its projection are represented in the computer. An object is stored in a two-dimensional array of numbers also called a *matrix*. Each of these number represents the density of one small square called *pixel*.

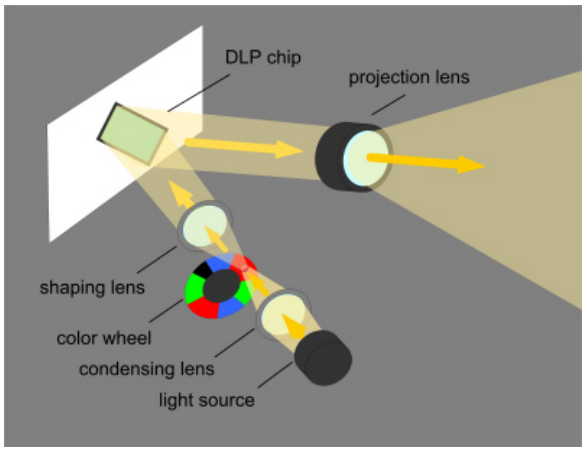


Figure 1.2: Single chip Digital Light Processing projection system layout

A projection on the other hand, is stored as a vector or list of numbers, each of which represents the total X-ray density in a narrow strip across the object. These numbers are also called a *ray* and the total density within the ray is called a *ray sum*. One major problem is that there is no relationship between the rays and the pixels.

In a Digital Light Processing (DLP) shown in Figure (1.2), light is focused through a rotating multi-color which are the red, green, and blue filter wheel. Then, let $\boldsymbol{x}$ denote the image to be reconstructed. The image is divided into $n \times n$ $(= N)$ dimensional vector, whose $[(u-1)n+v]$th denote the density in the pixel in the uth row and vth column. Now, let $\mathcal{G}$ be a matrix known only by the assumption whose (i,j)th element denotes the contribution of the jth pixel to the ith ray. Thus, $\mathcal{G}\boldsymbol{x}$ is a vector of the ray sums in the various rays. Practically, $\mathcal{G}$ is chosen arbitrarily, and this may contribute an error in the measured data.

1.4.1 Introduction to Inverse Problems

Inverse problems occur in a wide range of practical investigations where the variables of interest are indirectly measured by Magnetic Resonance Imaging (MRI), Computed Tomography (CT), or Positron Emission Tomography (PET) scanning in medical imaging. Many of these can be classified as function estimation or image processing problems, [Aykroyd (2015)]. In statistical field, key challenges include dealing with large number of unknown parameters compared to the amount of data

and the highly multicollinear nature of the design matrix. Linear inverse problem can be thought of as highly multivariate regression problems in which estimation is prioritized over prediction. Inverse problems can be ill-posed and ill-conditioned which makes estimation through least squares or maximum likelihood numerically unstable or even infeasible. However, a possible alternative can be to introduce a penalty term and use a penalized least squares or penalized maximum likelihood such as lasso regression [Tibshirani (1996)] or ridge regression [Hoerl & Kennard (1970)] for stable estimation. These shrinkage estimators are not appropriate as they would effectively introduce a bias towards zero.

1.4.2 General Bayesian Approach to Inverse problem

Consider the problem of finding $\boldsymbol{x} \in \mathbb{R}^N$ from the $\boldsymbol{y} \in \mathbb{R}^M$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are related by the equation below

$$\boldsymbol{y} = \mathcal{G}\boldsymbol{x} \tag{1.1}$$

$\boldsymbol{y}$ shall be referred to as the *observed data* and $\boldsymbol{x}$ as the *unknown* quantity. There are some reasons why this problem may be difficult. Typically, we focus on one of these reasons. One difficulty this thesis focus on is the case where $N = M$, which concerns the fact that often equation is perturbed by noise and so would consider the following equation in the chapter (6) of this thesis.

$$\boldsymbol{y} = \mathcal{G}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{1.2}$$

where we have denoted $\boldsymbol{\epsilon} \in \mathbb{R}^M$ as the *observational noise* which contaminates the observed data. Assume further that $\mathcal{G}$ maps $\mathbb{R}^M$ into a proper subset of itself, $\mathcal{G}'$ where $\mathcal{G}'$ represents the image of $\mathcal{G}$. There exists a unique inverse from $\mathcal{G}'$ into $\mathbb{R}^M$. Due to the noise, $\boldsymbol{y} \notin \mathcal{G}'$. So, simply inverting on the data $\boldsymbol{y}$ remains impossible. Moreover, the specific instances of the noise observation $\boldsymbol{\epsilon}$ in the data $\boldsymbol{y}$ is unknown. Then only the statistical properties of the noise is known which is always taken to be Gaussian with mean zero and constant variance. Thus, we cannot subtract $\boldsymbol{\epsilon}$ from the measured data $\boldsymbol{y}$ to get the noise-free data $\mathcal{G}(\boldsymbol{x}) \in \mathcal{G}'$. However, if $\boldsymbol{y} \in \mathcal{G}'$, the uncertainty posed by the noise $\boldsymbol{\epsilon}$ causes problems for the inversion. Probabilistic thinking enables us to overcome the difficulty of inversion discussed above to solve

the problem. We will treat $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{\epsilon}$ as random variables and determine the joint probability distributions of $(\boldsymbol{x}, \boldsymbol{y})$. The "solution" of the inversion problem is then defined as the probability of $\boldsymbol{x}$ given $\boldsymbol{y}$, denoted by $\boldsymbol{x}|\boldsymbol{y}$. This allows us to model the noise via its statistical properties without knowing the exact instances of the noise entering the given data. Moreover, it also enables us to specify *a priori* the form of solutions that we believe to be more likely, thereby enabling us to attach weights to multiple solutions which explain the data. This is the *Bayesian approach* to inverse problem.

We define a random variable $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^N \times \mathbb{R}^M$ as follows; $\boldsymbol{x} \in \mathbb{R}^N$ be a random variable with probability density $p(\boldsymbol{x})$. Equation (1.1) is defined as the likelihood $\boldsymbol{y}|\boldsymbol{x}$, where $\mathcal{G} : \mathbb{R}^N \to \mathbb{R}^M$ is measurable, and $\boldsymbol{\epsilon}$ is independent of $\boldsymbol{x}$ distributed according to measure $\mathbb{Q}_0$ with probability density $P(\epsilon)$. Then the random variable $\boldsymbol{y}|\boldsymbol{x}$ is simply found by $\mathbb{Q}_0$ and $\mathcal{G}$ to measure $\mathcal{Q}_{\boldsymbol{x}}$ with probability density $p(\boldsymbol{y} - \mathcal{G}(\boldsymbol{x}))$. Thus, $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^N \times \mathbb{R}^M$ is a random variable with probability density $p(\boldsymbol{y} - \mathcal{G}(\boldsymbol{x}))p(\boldsymbol{x})$. The Bayes' theorem allows us to calculate the distribution of the random variable $\boldsymbol{x}|\boldsymbol{y}$.

1.5 Probabilistic inference

Bayesian probability theory provides a language for beliefs representation and a calculus for manipulating these beliefs in a well organized manner. It is an extension of the formal theory of logic which is based on axioms that involve propositions that are true or false. The rules of probability theory are propositions which are plausible to being true of false and can be arrived at on the basis of just three *sine qua non*: (1) level of plausibility should be represented by real numbers; (2) common sense should have a qualitative measure with plausibilities; (3) different approaches to a conclusion should lead to the same result. Cox showed that plausibilities can be measured on any scale and it is possible to transform them onto the canonical scale of probabilities that sum to one. The product and sum rules of probability can be mathematically derived [Cox (1946)]. Statistical modeling problems involve large numbers of interacting random variables and it is often interesting to graphically represent these dependencies between the random variables. In particular, such graphical models are an important

tool for visualizing conditional independency relationships between variables. By rule of independence, a variable a is independent of variable b given variable c if and only if $p(a, b|c)$ can be written $p(a|c)p(b|c)$, By investigating conditional independence relationships, graphical models provide a background upon which it has been possible to derive efficient message-propagating algorithms for conditioning and marginalizing variables in the model given observation data [Pearl (1988); Lauritzen & Spiegelhalter (1988); Jensen (1996); Heckerman (1996); Cowell et al. (1999); Jordan (1999)]. So ideally, Bayesian methods should be applied to any situation where an inferences from data need to be made. These include classification or clustering task such as classifying cat and dog images, decision task like determining the next action of a robotic game. Despite these many desirable theoretical properties, Bayesian methods are less widely used in many interesting applications of artificial intelligence, especially those supported by deep learning [Goodfellow et al. (2016); LeCun et al. (2015); Schmidhuber (2015)].

Although the Bayesian approach maintains a posterior distribution of all possible settings of the desirable unknown factors, it also computes the marginal probability of the observations. For example, in discriminative supervised learning, one would define a conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$, which is also called *likelihood function* of $\boldsymbol{\theta}$. A concrete example for this would be to interpret $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ as outputting the probability of a configuration of $\boldsymbol{y}$ (e.g. a label or real value) by transforming the input $\boldsymbol{x}$ (an image, a text, etc.) through a simple model or deep learning model parameterized by $\boldsymbol{\theta}$. In any real-world observed data, the parameters $\boldsymbol{\theta}$ are unknown, but with prior knowledge of $p_0(\boldsymbol{\theta})$ about what value they might take. Then we collect the observations $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, and based on data we want to update our prior belief on the unknown parameters $\boldsymbol{\theta}$, for example: given $\mathcal{D}$, what is the most probable value of $\boldsymbol{\theta}$, and the likelihood of $\boldsymbol{\theta}$ to be set to a given value? Responding to these questions is the procedure of *inference* which is a procedure of deducing unknown properties given the observed data.

1.5.1 Emperical Bayes and hierarchical priors

It often makes sense to consider each parameter as coming from the same prior distribution when there are many common parameters in the vectors $\boldsymbol{\theta} = (\theta_1, ..., \theta_G)$.

For example, the prior specification of the means of each of the Gaussian components in a mixture model i.e. there is generally no a priori reason to expect any particular component to be different from another. The parameter prior is then formed from integrating w.r.t. a hyperprior with hyperparameter τ:

$$p(\boldsymbol{\theta}|\tau) = \int p(\tau) \prod_{g=1}^{G} p(\theta_g|\tau)\ d\tau \tag{1.3}$$

Therefore, each parameter is independent *given* the hyperparameter, although they are marginally independent. Hierarchical priors are useful even when applied only to a single parameter that often offers a more intuitive interpretation for the parameter's role. For example, the precision parameter τ of a Gaussian variable is often given a gamma prior, which contains two hyperparameters (α_τ, β_τ) and correspond to the shape and scale of the prior. Interpreting the marginal distribution of the variable in this generative context is often more intuitively appealing than simply enforcing a Student-t prior. Hierarchical priors are often designed using conjugate forms, both analytical ease and also readily expresses previous knowledge. For practical example, chapter (6) provide more details.

Hierarchical priors can be easily visualized using the graphical models which is shown in other chapters going forward. *Empirical Bayes* refers to the practice of optimizing the hyperparameters τ of the priors, in order to maximize the marginal likelihood of the data set $p(\boldsymbol{y}|\tau)$. By so doing, Bayesian learning can be viewed as optimizing marginal likelihood learning, where there are always distributions over the parameters. Moreover, the hyperparameters are optimized just as in maximum likelihood learning. One crucial limitation of this practice is that it ignores the uncertainty in the hyperparameters τ. An alternative approach could be to define priors over the hyperparameters and priors on the parameters of those priors etc. In this fashion, no parameters are actually ever fit for the data, and all predictions and inferences are based on the posterior distributions over the parameters.

1.5.2 Exact Bayesian inference

Bayesian statistics are particular in answering the questions on the unknown parameters $\boldsymbol{\theta}$ by computing the *posterior distribution*, or the *posterior belief* of $\boldsymbol{\theta}$ given $\mathcal{D}$, using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D})} \tag{1.4}$$

where $p(\mathcal{D}|\boldsymbol{\theta}) = \prod_n p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta})$ following the i.i.d. assumption. The importance of Bayes' rule is that it simultaneously handles *inference and modeling*. The model - the posterior distribution determined by the combination of the prior distribution and the likelihood. The inference by the value of the posterior estimates.

A core computational hurdle in Bayesian inference is *integration*. Using the product rule and sum rule of the probability distributions we derive the marginal distribution as follows

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{1.5}$$

here we have assumed that the random variable is continuous equivalently, this is the same for the discrete random variable which will be sum instead. If this integral is tractable, then the posterior distribution can be easily computed by Equation (1.5). To predict the label $\boldsymbol{y}^*$ on an unseen data $\boldsymbol{x}^*$, the predictive distribution is computed as follows

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \tag{1.6}$$

Equation (1.6) requires solving another integration problem. Since it is hard to visualize the posterior distribution in high dimensional space, we instead consider the statistics of the posterior such as the mean and variance as follows;

$$\boldsymbol{\mu} = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{1.7}$$

$$\boldsymbol{\Sigma} = \int (\boldsymbol{\theta} - \boldsymbol{\mu})(\boldsymbol{\theta} - \boldsymbol{\mu})^T p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{1.8}$$

which are both integration tasks. In particular, many tasks in Bayesian computation can be framed as computing an integral of some function $Q(\boldsymbol{\theta})$ against the posterior distribution:

$$\int Q(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{1.9}$$

and the goal of chapter (2) is dedicated to studying the different methods to computing integrals efficiently and pragmatically.

1.6 Practical Bayesian approaches

Bayes' rule provides a means of updating our belief system about the parameters from the prior to the posterior distribution in light of observed data. In theory, the posterior distribution encapsulates all the information deduced from the data about the parameters. This posterior is then used to make optimal decision or predictions, or model selection. These integrals are analytically intractable, and inaccessible to numerical integration techniques. However, the computations involve high dimensional integrals and for mixture models, the integrand has exponentially many modes. There are various ways we can tackle this problem. At one hand, one can be restricted to models and prior distributions that gives tractable posterior distributions and integral for the marginal likelihood and predictive densities. This is highly undesirable since it would lead to lose prior knowledge and modeling power. More realistically, we can approximate the exact answer.

1.6.1 Maximum a posteriori parameter estimates

The simplest approximation to the exact posterior distribution is to use a point estimate, such as Maximum a-Posteriori (MAP) parameter estimate,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}). \tag{1.10}$$

This chooses the model with highest posterior probability density i.e. the mode. However, this estimate is not completely Bayesian since the mode of the posterior may not be a better mirror of the posterior distribution, although it does contain some information from the prior. In particular, MAP models may likely give overconfident predictions, since all the posterior probability mass is contained in models that give poorer likelihood to the data. It might be reasonable to specify a collection of point estimates called *credible regions* instead of the MAP estimate. However, point estimates and credible regions have a limitation of *non-uniqueness* i.e., it is al-

ways possible to find one-to-one monotonic correspondent mapping of the parameters such that any particular parameter setting is at the mode of the posterior probability density in the mapped space provided that the probability has non-zero value under prior. This means that two models (identical priors and likelihood function) with different parameterizations will in general find different estimate, this is called *identifiability*. The key factor in the Bayesian approach is not just the use of a prior but the fact that all unknown variables are averaged over. This places a lesser importance on the choice of parameterization because the parameters will be integrated out.

1.6.2 Discussion of Identifiability

The convergence to Gaussian of the posterior holds if a model is identifiable. A model is unidentifiable if there are two or more parameters such that posteriors are equivalently related or if there is *degeneracy* in the parameter posterior.

Degeneracy arises in models with symmetries, where the assumption of a single mode in the posterior is incorrect. An example of symmetry is a model with a discrete latent variable $\boldsymbol{Z}_i$ having G possible settings i.e. indicator variable in a finite mixture model. These settings can be labeled in $G!$ ways since there is a latent variable. If the aliases are sufficiently distinct which correspond to unique peaks in the posterior due to large amount of data, the error in the approximation method can be corrected by multiplying the marginal likelihood by a factor of $G!$. However, it is difficult in practice to ascertain the level of separation of the aliases, and so a simple modification of this sort is impracticable. Estimating the *permanent* of the model are complicated and computationally burdensome [Barvinok (1999); Jerrum et al. (2001)].

Parameter degeneracy arises when there are some redundancies in the selection of parameterization of the model. For example, if a model has at least two parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$, and the prior over the parameter does not distinguish θ_1 from θ_2, then the posterior over $\boldsymbol{\theta}$ will contain an infinite number of distinct configurations of (θ_1, θ_2), all which give the same likelihood to the data. Consequently, the degeneracy causes marginal likelihood estimate irrelevant.

1.6.3 Monte Carlo methods

The large data limit approximations such as Laplace approximations are unfortunately limited in their ability to trade-off computational time to improve their accuracy. For example, the Hessian determinant that needs to be calculated exactly which costs $\mathcal{O}(nd^2)$ operations to find the Hessian and then $\mathcal{O}(d^3)$ to find its determinant. Laplace approximation may still be very inaccurate. Numerical integration methods is more accurate but is computationally intensive.

On the contrary, the Monte Carlo integration estimates the expectation of a function $\Gamma(\boldsymbol{\theta})$ under a probability distribution $f(\boldsymbol{\theta})$, by taking samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N} : \boldsymbol{\theta}^{(i)} \sim f(\boldsymbol{\theta})$. An unbiased estimate, $\hat{\Phi}$, of the expectation of $\Gamma(\boldsymbol{\theta})$ under $f(\boldsymbol{\theta})$, using N samples is given by:

$$\Phi = \int \Gamma(\boldsymbol{\theta}) f(\boldsymbol{\theta}) \simeq \hat{\Phi} = \frac{1}{N} \sum_{i=1}^{N} \Gamma(\boldsymbol{\theta}^{(i)}) d\boldsymbol{\theta} \tag{1.11}$$

Expectations such as the predictive density, the marginal likelihood, posterior distributions over the latent variables etc., can be obtained using such estimates. Most importantly, Monte Carlo method returns more accurate and reliable estimate which depends on large samples taken and the dimensionality of $\boldsymbol{\theta}$. Because in the continuous space, $Q(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})$ could never be computed at all locations, then one approach would be discretization and Monte Carlo methods. However, Monte Carlo approaches assume that drawing samples from the posterior distribution is easy, which is again intractable in most cases. Statisticians have applied advanced drawing schemes such as importance sampling, rejection sampling and MCMC [Gelman et al. (2014)] to draw samples from the posterior distribution. Moreover, these methods require a considerable number of samples in high dimensional space. These methods and the simulation time for MCMC can be prohibitively long due to a slow mixing.

1.6.4 Importance sampling

In situations where the distribution $f(x)$ is difficult to sample from, one can sample from a correlated distribution called an *auxiliary distribution* $g(x)$ and then correct for this by weighing the samples accordingly. This is called *Importance sampling* and it constructs the following estimator using N samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N}$, generated such that

each $\boldsymbol{\theta}^{(i)} \sim g(\boldsymbol{\theta})$:

$$\Phi = \int g(\boldsymbol{\theta}) \, \frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \, \Gamma(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \simeq \hat{\Phi} = \frac{1}{N} \sum_{i=1}^{N} w^{(i)} \Gamma(\boldsymbol{\theta}^{(i)}), \tag{1.12}$$

where $w^{(i)} = \frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}$ are called *importance weights*. Unfortunately, this estimate is biased as it is the ratio of two estimates, and the ratio of two unbiased estimates does not necessarily in general result in an unbiased estimate of the ratio. Although, importance sampling is simple, the estimate $\hat{\Phi}$ can often have a very high variance. Having the integration task at hand, the first thing that comes to mind is how possible they could be computed in an exact form or the question to ask is how tractable are they? However, very unfortunately, for a vast number of integrands and distributions, the integral in Equation (1.5) does not have a closed form or exhibit an analytical form. In general, the intractability of the marginal probability makes the posterior and predictive distributions intractable. Moreover, instead of finding exact form of the integral, many mathematicians have their research career in an alternative method such as a *numerical integration*.

1.6.5 Rejection sampling

A related method to importance sampling is *rejection sampling* which avoids the use of a set of weights ${w^{(i)}}_{i=1}^{N}$ by stochastically deciding whether or not to include each sample from $g(\boldsymbol{\theta})$. The procedure requires the existence of a constant r such that $rg(\boldsymbol{\theta}) > f(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, that means $rg(\boldsymbol{\theta})$ envelopes the probability density $f(\boldsymbol{\theta})$. Samples are indirectly obtained from $f(\boldsymbol{\theta})$ by sampling from $g(\boldsymbol{\theta})$, and then a condition which is either accepting or rejecting each stochastically is used based on the ratio of its density under $f(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta})$. That is to say for each sample is drawn uniform distribution $u^{(i)} \sim U(0, 1)$ and is accepted only if

$$f(\boldsymbol{\theta}^{(i)}) > u^{(i)} r g(\boldsymbol{\theta}^{(i)}). \tag{1.13}$$

However, this becomes impractical in high dimensions and with complex functions since it is difficult to find a simple choice of $g(\boldsymbol{\theta})$ such that r is small enough to allow the rejection rate to remain reasonable across the whole space. To overcome

the limitations of rejection sampling, it is important to focus on tightness of bound between $f(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta})$ when $f(\boldsymbol{\theta})$ is log-concave.

1.6.6 Markov Chain Monte Carlo

MCMC methods reviewed in Neal (1992) has been used to generate a chain of correlated samples, starting from $\boldsymbol{\theta}^{(1)}$ such that the next sample is a stochastic function of the previous sample: $\boldsymbol{\theta}^{(i)} = p(\boldsymbol{\theta}^{(i-1)})$ where $p(\boldsymbol{\theta}', \boldsymbol{\theta})$ is the probability of transition from $\boldsymbol{\theta}'$ to $\boldsymbol{\theta}$. If p has $f(\boldsymbol{\theta})$ as its stationary (equilibrium) distribution, i.e. $f(\boldsymbol{\theta}) = \int f(\boldsymbol{\theta}')p(\boldsymbol{\theta}', \boldsymbol{\theta})d\boldsymbol{\theta}$, then the set $\{\boldsymbol{\theta}\}_{i=1}^{N}$ is used to obtain an unbiased estimate of Φ in the limit of a large samples. Since the set of samples have to be drawn from the equilibrium then, it is advisable to discard all samples drawn at the beginning of the chain, this is called *burn-in*. The importance of this is reduce the *autocorrelation* among the chains. $\mathcal{P}$ in general is implemented using a proposal density $\boldsymbol{\theta}^{(i)} \sim g(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ about the previous sample. The probability of accepting the proposal needs to take into account the probability of a reverse transition, this is to ensure *reversibility* of the Markov chain. This method gives rise to the Metropolis-Hastings [Metropolis et al. (1953); Hastings (1970)] acceptance function $a(.,.)$:

$$a(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}) = \frac{f(\boldsymbol{\theta}^{(i)})g(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\theta}^{(i)})}{f(\boldsymbol{\theta}^{(i-1)})g(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)})} \tag{1.14}$$

If the acceptance function in Equation (1.14) is greater than one, i.e. $a(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}) \geq 1$ the sample is accepted, otherwise it is accepted according to the probability $a(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)})$. Although, MCMC is guaranteed to yield exact estimates in the limit of a large number samples even for a well-designed procedures, the number of samples required for acceptance estimates can be enormously large. Finally, Monte Carlo is a purely frequentist procedure and according to Hastings (1987) it is fundamentally unsound.

1.6.7 Approximate Bayesian inference as optimization

Here comes an outstanding idea of *approximate inference* that finds another distribution $q(\boldsymbol{\theta})$ that is in the same exponential family with the exact posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ and can make the computation of the integral $\int Q(\boldsymbol{\theta})q(\boldsymbol{\theta})$ relatively easier by concurrently reducing the minimal approximation error to the exact integral we want

to compute. The knowledge of the functional form Q can be used to form a class of candidate distributions $\mathcal{Q}$ in which integrating Q w.r.t. any $q \in \mathcal{Q}$ is analytically tractable or computable with numerical methods. Then the major goal is to find the *optimal* solution q distribution in $\mathcal{Q}$ such that the q integral is the most probable approximation to the exact one. So, *approximate inference* presents the integration problem of Bayesian inference as an *optimization* goal. For example, an approach for fitting the q distribution would be to minimize a distance, divergence or discrepancy measure between the approximate distribution and the exact posterior distribution.

$$q^*(\boldsymbol{\theta}) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \, \mathrm{KL}[q(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\mathcal{D})] \tag{1.15}$$

Equation (1.15) is called *Kullback Leibler (KL) divergence* which is the popular choice for the divergence measure [Kullback (1959); Kullback & Leibler (1951)]. This is widely used in *variational inference* algorithm [Beal (2003); Ghahramani & Beal (2000); Jordan et al. (1999)]. In general, an optimization objective function $\mathcal{F}$ is designed to obtain an accurate approximation:

$$q^*(\boldsymbol{\theta}) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \, \mathcal{F}\Big(q(\boldsymbol{\theta}); p(\boldsymbol{\theta}|\mathcal{D})\Big), \tag{1.16}$$

This objective function $\mathcal{F}$ is formulated such that $\mathcal{F}^*$ can represent an accurate approximation to the logarithm of the marginal distribution, or *model evidence* $\log p(\mathcal{D})$ at the optimum. All these methods are thoroughly discussed in Chapter (2). The Bayesian prediction distribution in Equation (1.6) is computed at the prediction time once the approximate posterior q is obtained:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) \approx \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}) q(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{1.17}$$

1.7 Summary of the remaining Chapters

Chapter (2) discusses the background study on the family of variational Bayesian algorithms. It investigates the variants of EM algorithms with different maximization techniques such as OLS, IRLS, and SGD. It discusses an alternative to variational Bayesian algorithm which is Expectation propagation algorithm. The message passing

EP-MCMC and EP-ADMM are also discussed. This is then applied to a hierarchical Bayesian inverse problem in Chapter (6).

Chapter (3) discusses the general idea of Cluster weighted models. Diving through from the finite mixture model, we give some historical evolution of the finite mixture model. In the subsequent sections, the history about the method of estimation is given with more emphases on *EM* algorithm. Afterwards, we study the CWMs in high-dimensional and introduce a dimensionality reduction technique called tSNE. We applied the CWM-tSNE on large data and aim to discover the hidden structure of the data.

Chapter (4) proposes a new family of generalized cluster weighted model called *Multinomial CWMs*. It extends the binomial CWMs for the binary response variable in twofold. Firstly, MCWM allows for the possible nonlinear dependencies in the mixture components by considering a multinomial logit regression or softmax regression for multi-class. Secondly, it considers multinomial distribution for the conditional distribution of the response variable given the covariates. it investigates the conditions under which the proposed model is identifiable. Conventionally, maximum likelihood estimations are derived using the Expectation Maximization algorithm for cluster weighted models. However, EM algorithm is known for its slow-to-convergence nature and inability to scale to large dataset due to its slow nature. To overcome this drawback in EM and to avoid the problem matrix inversion of the model arising from the EM algorithm with an iteratively re-weighted least squares (EM-IRLS), we use the Expectation Maximization with a Stochastic Gradient Descent (EM-SGD). Model selection is carried out using the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and Integrated Completed Likelihood (ICL) and other five variants. ARI variants such as Rand Index, Huberts and Arabie's (HA), Fowlkes and Mallow's (FM), Morey and Agresti's (MA), and Jaccard (JA) are considered as a different measure of accuracy. The clustering performance of the proposed model is investigated through simulation and real data sets. Considering different datasets,

multinomial CWM shows excellent clustering results via performance measures such as Accuracy and Area under the ROC curve.

Chapter (5) proposes another model into the family of Cluster Weighted Models (CWMs) called *Zero-Inflated Cluster Weighted Model* (ZIPCWM). It extends Poisson cluster weighted models and other mixture models. To estimate the parameter of the models, an EM algorithm with an iteratively re-weighted least squares is proposed. The ZIPCWM is applied to real data which accounts for excess zeros of over 40%. We explore the classification performance of ZIPCWM, Fixed Zero-Inflation Poisson Mixture Model (FZIP), and Poisson Cluster Weighted Model (PCWM) on the data. In conclusion, ZIPCWM outperforms both PCWM and FZIP models.

Chapter (6) focuses on solving the problem of a deterministic algorithm called *Expectation Propagation* algorithm. It proposes an easy-to-use reconstruction method based on Expectation Propagation (EP) techniques. In order to address the intractability of the normalizing factor in EP, this method incorporates the Monte Carlo methods, MCMC, and Alternating Directional Method of Multiplier (ADMM) algorithm into EP method. The advantages of the proposed technique include stability derived from a stochastic search and a flexibility to hierarchical models. It demonstrates the approach on complex Bayesian models for image reconstruction. Our technique is applied to images from Gamma-camera scans. The experiment focuses on image reconstruction from a noisy image. It compares the results from the proposed method with that produced by MCMC.

Chapter 7 concludes the thesis with a discussion and summary of the main contributions of the thesis.

Chapter 2

Family of Variational Bayesian Theory

2.1 Introduction

To open the floor on the discussion of the family of variational methods, I provide a condensed introduction to three classes of well-known approximation inference techniques such as Expectation Maximization (EM), Variational Bayes (VB), and Expectation Propagation (EP). I will first start by discussing statistical divergence measures upon whose neck they all lean. Then I will provide the extension of two methods in details.

2.2 Statistical divergence measures

Many approximate inference algorithms measure the approximation quality by considering the *divergence* between the exact posterior and the approximation posterior. In this thesis, we will manly focus on the case where both the exact and the approximation are expressed by probability distributions. Before we dig further, we briefly discuss the definition of the *Probability Density Function* (PDF) as a prerequisite. Denote the measurable space $(\mathcal{X}, \mathcal{A})$, where $\mathcal{X}$ is the sample space of the random variable $\boldsymbol{x}$ of interest, and $\mathcal{A}$ is a pre-defined σ-algebra on $\mathcal{X}$. A *probability distribution* P is a measure defined on $\mathcal{A}$ such that $P(\mathcal{X}) = 1$. Also, we assume there exists

a dominating or reference measure μ on $\mathcal{A}$ such that, for a probability distribution P defined on $\mathcal{A}$, we define its *probability density function* p by $dP = pd\mu$ where for discrete case, the PDF is referred to as *probability mass function* (PMF). For simplicity, we will work with the sample space $\Theta = \mathbb{R}^D$, the σ-algebra $\mathcal{A} = \{S : S \subset \mathbb{R}^D\}$, and the reference measure $d\mu = d\boldsymbol{\theta}$. Finally, $\mathcal{P}$ denote the space of the PDFs such that any probability distribution P defined on $\mathcal{A}$ has its PDF $p \in \mathcal{P}$. The next section discuses the former definition of divergence.

Definition 2.1 (Divergence): *Given a set of probability density functions $\mathcal{P}$ for a random variable $\boldsymbol{\theta}$, a divergence on $\mathcal{P}$ is defined as a function $\boldsymbol{D}[.||.] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ such that $\boldsymbol{D}[p||q] \geq 0$ for all $p, q \in \mathcal{P}$, and $\boldsymbol{D}[p||q] = 0$ iff $p = q$.*

The above definition is much weaker than that of a *distance* such as l_2-norm, since it does not need to satisfy either symmetry in arguments or the triangle inequality. Hence there are many types of divergences, and this section discusses some of the popular choices. We kick off from the well-known Kullback-Leibler (KL) divergence and its properties and applications. Then in general, we review α-divergences and focus more on the KL divergence as the main divergence tools for the algorithms developed in this thesis.

2.2.1 Kullback-Leibler (KL) divergence

Kullback-Leibler divergence is one of the most widely used divergence measures, both in approximation problem in Bayesian inference and in machine learning, statistics, and information theory, [Kullback & Leibler (1951); Kullback (1959)].

Definition 2.2 *(Kullback-Leibler Divergence): The Kullback-Leibler (KL) divergence on $\mathcal{P}$ is defined as a function $KL[.||.] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ with the following form*

$$\mathrm{KL}[p||q] = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}, \; p, q \in \mathcal{P}, \tag{2.1}$$

It is easy to check the validity of the above definition as a divergence. By Jensen's inequality, Equation (2.1) is always non-negative and it attains zero iff. $p = q$. Also, it is apparent that KL divergence is asymmetric, i.e. $\mathrm{KL}[p||q] \neq \mathrm{KL}[q||p]$. Historically, when used in approximate inference field, the $\mathrm{KL}[p||q]$ is referred to as *inclusive* KL divergence, and $\mathrm{KL}[q||p]$ is referred as *exclusive* KL divergence. These names emanate from the idea that fitting q to p by minimizing these two KL divergence returns results of distinct behavior. Fitting q to p by minimizing $\mathrm{KL}[p||q]$, means that KL divergence assigns *high* probability mass of q to the location where p has positive mass, thus the name "inclusive" KL. This property is referred to as *mass-covering*. For example, in a region $S \in \boldsymbol{\Theta}$, the case that $q(\boldsymbol{\theta}) > 0$ but $p(\boldsymbol{\theta}) = 0$ would make the integrand in Equation (2.1) zero. In contrast, if $p(\boldsymbol{\theta}) > 0$ but $q(\boldsymbol{\theta}) = 0$, this would mean the integrand is infinity and the cost of missing a region with positive p mass is extremely high. Similarly, fitting q to p by minimizing $\mathrm{KL}[q||p]$ assigns *low* probability mass of q to the location where p is very small which means "exclusive" KL. For example, $S \in \boldsymbol{\Theta}$ that has $q(\boldsymbol{\theta}) > 0$ but $p(\boldsymbol{\theta}) = 0$ for $\boldsymbol{\theta} \in S$, then this makes the integrand in Equation (2.1) infinity, thus the KL divergence assigns an extremely cost to q. On the contrary, if $p(\boldsymbol{\theta}) > 0$ but $q(\boldsymbol{\theta}) = 0$, the integrand restricted to the subset S is zero, which means that the cost of missing a region with positive p mass is much lower. This is referred to as "zero-forcing", when the approximate distribution q is restricted to be unimodal.

Later we will see how these two KL divergence form the basis for widely used approximate algorithms such as expectation maximization algorithm [Dempster et al. (1977)], expectation propagation [Minka (2001)], and variational Bayes [Beal (2003); Jordan et al. (1999)]. In EM algorithm, it is obvious that maximizing the MLE is equivalent to minimizing a KL divergence.

2.2.2 Alpha divergences

Historically, just after a year of the proposal of KL-divergence, a test statistic for the likelihood-ratio test was introduced by [Chernoff (1952)]. The test statistic was linked to a divergence measure that is computed by the infimum of an integral. The

integral which has been referred to as the *Chernoff* α-coefficient is as follows

$$\int p(\boldsymbol{\theta})^{\alpha} q(\boldsymbol{\theta})^{1-\alpha} d\boldsymbol{\theta}, \qquad \alpha \in (0,1), \tag{2.2}$$

It was argued by a mathematician in 1961 that, Shannon entropy can be further generalized to many interesting cases by removing the additivity requirement [Rényi (1961)]. He then proposed one of such entropy definitions, and then characterized the induced mutual information and relative entropy measures using his version of α-divergence called *Renyi entropy* and *Renyi divergence*, respectively.

In the $20th$ century, differential geometry that studies the geometric properties of the manifold was introduced to statistics [Amari (1985); Efron (1975); Efron (1978)]. This is obtained by mapping $\mathcal{P}$ to the parameter space $\boldsymbol{\Theta}$. In particular, geometric properties of *exponential family* distribution was the main focus and the corresponding divergences that reflect these features. Following this path, mathematician Shun-ichi Amari introduced his version of α-divergence [Amari (1982); Amari (1985)], by generalizing the application of Chernoff α-coefficient to $\alpha \in \mathbb{R}$.

Definition 2.3 *(Amari's α-divergence): Amari's α-divergence $\boldsymbol{D}_{\alpha}^{A}[.||.] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, parameterized by $\alpha \in \{\alpha : \mathcal{D}_{\alpha}^{A}[p||q] < +\infty\}$, is defined as*

$$\boldsymbol{D}_{\alpha}^{A}[p||q] = \frac{4}{1-\alpha^2}\left(1 - \int p(\boldsymbol{\theta})^{\frac{1+\alpha}{2}} q(\boldsymbol{\theta})^{\frac{1-\alpha}{2}}\right). \tag{2.3}$$

The limit of Equation (2.3) as $\alpha \to 1$ is $\mathrm{KL}[p||q]$ and the limit of Equation (2.3) as $\alpha \to -1$ is $\mathrm{KL}[q||p]$. Amari used α-divergence to study the geometry of distribution manifolds, and as claimed in Amari (2009). Amari's definition is the only divergence that belongs to both f-divergence [Csiszár (1963)] and Bregman divergences [Bregman (1967)].

2.2.3 Renyi's Alpha divergence

Renyi's α-divergence measures the "closeness" of two distributions p and q on a random variable $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. It is defined on $\{\alpha : \alpha \neq 1, \alpha > 0\}$ as follows

$$\boldsymbol{D}_{\alpha}^{R}[p||q] = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{\theta})^{\alpha} q(\boldsymbol{\theta})^{1-\alpha} d\boldsymbol{\theta}. \tag{2.4}$$

In Equation (2.4), $|\boldsymbol{D}_{\alpha}^{R}[p||q]| < +\infty$ is useful to rewrite the divergence as the expectation under p or q. The definition is extended to $\alpha = 0, 1, +\infty$ by continuity. When $\alpha \to 1$, the Kullback-Leibler (KL) divergence is recovered, which plays a crucial role in machine learning and information theory.

Table 2.1: Special cases in the Renyi divergence family

α	Definition	Notes
$\alpha \to 1$	$\int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}$	*Kullback-Leibler (KL) divergence* used for KL$[q\|\|p]$ in VB and KL$[p\|\|q]$ in EP.
$\alpha \leftarrow 0$	$-\log \int_{p(\boldsymbol{\theta})>0} q(\boldsymbol{\theta}) d\boldsymbol{\theta}$	Zero when $\sup(q) \subseteq \sup(p)$
$\alpha = 0.5$	$-2\log(1 - H^2[p\|\|q])$	Square *Hellinger distance*
$\alpha = 2$	$-\log(1 - \chi^2[p\|\|q])$	proportional to the χ^2 divergence
$\alpha \leftarrow +\infty$	$\log max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{p(\boldsymbol{\theta})}{\boldsymbol{\theta}}$	*worst-case regret used in minimum description length principle*[Grunwald, 2007]

Besides the KL divergence, there are other choices of divergences for approximate inference presented in Table (2.1). Some Properties of the α-divergence are given below;

Proposition 2.1

(Monotonicity): Renyi's α-divergence extended to negative α is ***continuous*** *and* ***non-decreasing*** *on $\alpha \in \{\alpha : -\infty < \boldsymbol{D}_{\alpha}^{R} < +\infty\}$.*

Proposition 2.2

(Skew symmetry): For $\alpha \notin \{0,1\}$ *and* $\boldsymbol{D}_{\alpha}^{R}[p||q] = \frac{\alpha}{1-\alpha}\boldsymbol{D}_{1-\alpha}^{R}[q||p]$. *This implies* $\boldsymbol{D}_{\alpha}^{R}[p||q] \leq 0$ *for* $\alpha < 0$. *For the limiting case* $\boldsymbol{D}_{-\infty}^{R}[p||q] = -\boldsymbol{D}_{+\infty}^{R}[q||p]$. A current active research is how to choose a divergence in this rich family to obtain optimal solution for a particular application.

2.3 Expectation Maximization algorithm

The EM algorithm is an iterative maximum-likelihood estimator and is typically used when one is dealing with *incomplete data* or when the likelihood function involves *latent variables.* However, the distinction of the two cases is more of interpretation issue, since latent variable can be thought of as an unobserved data, thus leading to incomplete data. Therefore, it is possible to use the EM algorithm by introducing a pseudo variables which are simply declared as unobserved. The EM algorithm was discovered and used independently by several different researchers, but it was first described fully by [Dempster et al. (1977)], who also coined the term "EM algorithm".

One important problem that motivated this algorithm was the parameter estimation of Gaussian Mixture Models [McLachlan & Peel (2000)], since Cluster Weighted Models are closely related to this problem, it has also been used for parameter estimation. Since Dempster's paper, a huge amount of material was published which further investigated and used the algorithm for various purposes. It is important for training of hidden Markov models, especially for speech recognition, pattern recognition, neural network training.

2.3.1 General formulation

For describing the EM algorithm in a general fashion, we consider a parametric density function $p(x|\boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ are the parameters; in the case of CWMs, these parameters are the cluster weights, the location and variances of the clusters. The random variable X is assumed to be i.i.d according to this distribution, and $\boldsymbol{\mathcal{D}} = \{x_1, ..., x_N\}$ is a data set which is also a realization of this random variable. The likelihood function is given

by

$$L(\mathbf{\Theta}|X,y) = p(X,y|\mathbf{\Theta}) = \sum_{i=1}^{N} p(x_i,y_i|\mathbf{\Theta}). \tag{2.5}$$

To obtain the value of the estimates of the parameters $\mathbf{\Theta}$, the common appraoch is to use those parameters that maximize the likelihood i.e. those parameters that fulfills the following equation

$$\hat{\mathbf{\Theta}}_{ML} = \underset{\Theta}{\operatorname{argmax}}\, L(\mathbf{\Theta}|X,y). \tag{2.6}$$

In many application cases, it is easier to calculate this maximization by using the log-likelihood $\mathcal{L}(\Theta|\mathbf{x})$, which shall be used in the following. Until now, computing this maximization analytically is often intractable or even impossible, so resorting to approximation procedures is the best choice.

Customarily, the EM algorithm is peculiar to incomplete data. This covers two different scenarios such as incomplete data resulting from a loss of information or expensiveness of observing some data or due to technical difficulties or other limitations of observation processes. The second scenario is to simplify the maximization of the likelihood by introducing a hidden variables. The latter is common in the cases of mixture and cluster weighted models.

The basic strategy of EM is to first formulate the complete data problem, given a hypothetical complete data set $\mathcal{D} = \{x, ..., x_N, y_1, , ..., y_N, z_1, ..., z_G\}$, where the z_i denote the latent or hidden variables. The complete log-likelihood is given by

$$\mathcal{L}(\mathbf{\Theta}|X,Y,Z) = \log p(X,Y,Z|\mathbf{\Theta}), \tag{2.7}$$

and we further decompose the joint probability according to Baye's rule into

$$p(X,Y,Z|\mathbf{\Theta}) = p(Z|X,Y,\Theta)p(X,Y|\mathbf{\Theta}). \tag{2.8}$$

Since hidden variables z_i are realizations of the random variable Z, the complete log-likelihood $\mathcal{L}_C$ is also a random variable, and we can calculate its expected value with respect to Z, given the observed data $\{x_i\}$ and a current parameter estimate $\mathbf{\Theta}^{(k)}$.

This expectation is usually written as the function Q

$$Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k)}) = E\Big[\log p(X, Y, Z|\boldsymbol{\Theta}) X, Y, \boldsymbol{\Theta}^{(k)} \Big]. \tag{2.9}$$

This is the *Expectation step* (E-step) of the algorithm. It is important to note that Z is a random variable and the $\boldsymbol{\Theta}$ can be a random variable in the case of variational Bayes algorithm, but in EM algorithm $\boldsymbol{\Theta}^{(k)}$ and X are constant. While at each iteration, X remains the same, the $\boldsymbol{\Theta}^{(k)}$ changes with the iteration k.

In the *Maximization step* (M-step), the recently calculated expectation $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k)})$ is maximized with respect to $\boldsymbol{\Theta}$, the result being our parameter estimate for the next iteration:

$$\boldsymbol{\Theta}^{(k+1)} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k)}) \tag{2.10}$$

This assures us that we will always climb uphill at every iteration and therefore maximizing the complete log-likelihood. However, if it is not possible then it is possible to find $\boldsymbol{\Theta}^{(k+1)}$ with the largest likelihood than the previous one. This idea is guaranteed to find a local maximum of Q. In every iteration, one E- and M-step are performed alternatively. The goal is to determine if these iteration will lead to convergence, and if so, it will converge to ML estimator.

Let Z take a value from the space of all possible values $\mathcal{Z}$, so that

$$\sum_{Z \in \mathcal{Z}} p(Z|X, Y, \boldsymbol{\Theta}^{(k)}) = 1, \tag{2.11}$$

and hence it holds that

$$\log p(X, Y|\boldsymbol{\Theta}) = \sum_{Z \in \mathcal{Z}} p(Z|X, Y, \boldsymbol{\Theta}^{(k)}) \log \left[\frac{p(X, Y, Z|\boldsymbol{\Theta})}{p(Z|X, Y, \boldsymbol{\Theta})} \right] \tag{2.12}$$

$$= \sum_{Z \in \mathcal{Z}} p(Z|X, Y|\boldsymbol{\Theta}^{(k)}) \log p(X, Y) \tag{2.13}$$

using the Q-function in Equation (2.9), we can see that the first term in this equation is $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(k)})$.

Therefore from Equation (2.12), we can see that

$$\log \frac{p(X,Y|\boldsymbol{\Theta})}{p(X,Y|\boldsymbol{\Theta}^{(k)})} = \log p(X,Y|\boldsymbol{\Theta}) - \log p(X,Y|\boldsymbol{\Theta}^{(k)}), \tag{2.14}$$

$$Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(k)}) - Q(\boldsymbol{\Theta}^{(k)},\boldsymbol{\Theta}^{(k)}) + \sum_{Z\in\mathcal{Z}} p(Z|X,Y,\boldsymbol{\Theta}^{(k)}) \log \frac{p(Z|X,Y,\boldsymbol{\Theta}^{(k)})}{p(Z|X,Y,\boldsymbol{\Theta})}, \tag{2.15}$$

The last term in Equation (2.15) is the Kullback-Leibler divergence between two densities in the fraction, which is not symmetric that is $\mathrm{KL}(p||q) \neq \mathrm{KL}(q||p)$ and by definition it is always positive. Therefore, it holds that

$$\log p(X,Y|\boldsymbol{\Theta}) - \log p(X,Y|\boldsymbol{\Theta}^{(k)}) \geq Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(k)}) - Q(\boldsymbol{\Theta}^{(k)},\boldsymbol{\Theta}^{(k)}). \tag{2.16}$$

Through M-step, we can now calculate $\boldsymbol{\Theta}^{(k+1)}$ i.e. through the maximization of the Q function

$$\boldsymbol{\Theta}^{(k+1)} \leftarrow \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(k)}). \tag{2.17}$$

Substituting Equation (2.17) into Equation (2.16) yields,

$$\log p(X,Y|\boldsymbol{\Theta}^{(k+1)}) - \log p(X,Y|\boldsymbol{\Theta}^{(k)}) \geq Q(\boldsymbol{\Theta}^{(k+1)},\boldsymbol{\Theta}^{(k)}) - Q(\boldsymbol{\Theta}^{(k)},\boldsymbol{\Theta}^{(k)}) \geq 0 \tag{2.18}$$

and therefore,

$$\log p(X,Y|\boldsymbol{\Theta}^{(k+1)}) - \log p(X,Y|\boldsymbol{\Theta}^{(k)}). \tag{2.19}$$

This means that, it is guaranteed that log-likelihood increases at each iteration, until the stopping criteria is satisfied when there is no or little change in the difference which implies that a fixed point is reached. From Equation (2.16), it is clear that every maximum likelihood parameter $\boldsymbol{\Theta}^{ML}$ is a fixed point of the iteration, and the sequence of parameter estimates $\boldsymbol{\Theta}^{(0)}, \boldsymbol{\Theta}^{(1)}, ..., \boldsymbol{\Theta}^{(k)}$ gives rise to a non-decreasing sequence $\mathcal{L}(\boldsymbol{\Theta}^{(0)}), \mathcal{L}(\boldsymbol{\Theta}^{(1)}), ..., \mathcal{L}(\boldsymbol{\Theta}^{(k)})$ which must converge as $k \to \infty$. In rare cases, it is possible to reach a saddle point and even a minimum of the likelihood [Wu (1983)].

2.3.2 Algorithm for General EM

Algorithm 1 The General EM Algorithm

1: Given a probability model with a set of observed variables $\mathcal{X}$, a set of unobserved latent variables $\mathcal{Z}$, and a vector of unknown parameters $\boldsymbol{\Theta}$, the goal is to maximize the log-likelihood w.r.t. the parameters $\boldsymbol{\Theta}$.

2: Start with an initial value for the parameter $\boldsymbol{\Theta}^{(0)}$ and compute the initial log-likelihood $\log p(\mathcal{X}|\boldsymbol{\Theta}^{(0)})$.

3: **E-step**: Evaluate

$$q^{(k)} = \underset{q}{\operatorname{argmax}}\, \mathcal{L}(q, \boldsymbol{\Theta}^{(k)}) = p(\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{\Theta}^{(k)})$$

4: **M-step**: Evaluate

$$\boldsymbol{\Theta}^{(k+1)} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(k)})$$

5: Compute the log-likelihood $\log p(\mathcal{X}|\boldsymbol{\Theta}^{(k+1)})$ and check for convergence of the algorithm. If the convergence criterion is satisfied return the final parameters. Otherwise, repeat steps $2-4$.

However, EM algorithm is a steepest ascent method which is only guaranteed to reach a local maximum of the likelihood. This is dependent on the initial parameterization $\boldsymbol{\Theta}^{(0)}$ which charted the course of the algorithm, i.e. set the route for the algorithm in hunting for the best parameter that maximizes the likelihood. Therefore, it is advisable to execute the algorithm with several starting values and compare the performance of the resulting parameters with some model selection procedure.

2.3.3 Convergence Criterion

The general convergence of the algorithm is discussed in detail. Having seen that the goal of both the E and the M steps are to maximize the log-likelihood $\mathcal{L}(q, \boldsymbol{\Theta})$, it is easy to see that after E-step,

$$l(\boldsymbol{\Theta}^{(k)}) = \mathcal{L}(q^{(k)}, \boldsymbol{\Theta}^{(k)}) \geq \mathcal{L}(q^{(k-1)}, \boldsymbol{\Theta}^{(k)}) \tag{2.20}$$

and that after the M-step,

$$\mathcal{L}(q^{(k)}, \boldsymbol{\Theta}^{(k+1)}) \geq \mathcal{L}(q^{(k)}, \boldsymbol{\Theta}^{(k)}). \tag{2.21}$$

Combining these two inequalities together, we obtain

$$\mathcal{L}(q^{(k+1)}, \boldsymbol{\Theta}^{(k+1)}) \geq \mathcal{L}(q^{(k)}, \boldsymbol{\Theta}^{(k)})$$

and

$$\log p(\mathcal{X}|\boldsymbol{\Theta}^{(k+1)}) \geq \mathcal{L}(q^{(k)}, \boldsymbol{\Theta}^{(k)}).$$

It is also clear that each EM iteration increases the lower bound on the log-likelihood function and will change the model parameters in such a way as to increase the actual log-likelihood. This is guaranteed to reach a local maximum of the log-likelihood function.

2.3.4 Maximum A posterior estimation

The EM algorithm can be used to find MAP solutions for models in which a prior $p(\boldsymbol{\Theta})$ over the parameters $\boldsymbol{\Theta}$ is employed. By decomposing the log-likelihood, we have

$$\log p(\boldsymbol{\Theta}|\mathcal{X}) = \log p(\mathcal{X}|\boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta}) - \log p(\mathcal{X})$$

$$\geq \mathcal{L}(q, \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta}) - \log p(\mathcal{X}) \tag{2.22}$$

where $\log p(\mathcal{X})$ is a constant. Again we can maximize the right-hand side alternatively with respect to q and $\boldsymbol{\Theta}$. In this case the E-step remains the same as in the maximum likelihood case as q only appears in $\mathcal{L}(q, \boldsymbol{\Theta})$, whereas the M-step the quantity to be maximized is given by $\mathcal{L}(q, \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta})$, which typically requires only a small modification to the standard maximum likelihood M-step.

2.3.5 Choosing the optimal number of components

When the primary goal is to identify the number of components G, testing for the number of components is very crucial. However, it is somewhat difficult in practice which still remains completely unsolved. When working with the mixture model,

two main purposes come to mind. One is to provide an appealing semi-parametric framework in which to model unknown distributional shapes [Escobar & West (1995); Robert (1996); Roeder & Wasserman (1997)]. The other purpose is to use the mixture model to provide a model-based clustering. In both situations, we need to answer the question of how many components is to be included in the mixture. In the previous studies of density estimation, the commonly used information criteria are the AIC and BIC which appear to be adequate for choosing the number of components G for a density estimation. Under mild conditions, Leroux (1992a) established that certain penalized log-likelihood criteria such as AIC and BIC do not underestimate the true the number of components asymptotically.

2.3.6 Akaike's Information Criteria

Akaike's Information criteria proposed by Akaike (1973) and Akaike (1974)] is used to select the model that minimizes

$$AIC = -2 \log l(\hat{\boldsymbol{\Theta}}) + 2m \tag{2.23}$$

where m is the total number of free parameters in the model. AIC has been widely used to assess the order of a mixture model. However, Koehler & Murphee (1988) argued that AIC is inconsistent and tends to overfit models. In mixture model, it is observed that AIC tends to overestimate the correct number of components [Koehler & Murphee (1993); Celeux & Soromenho (1996)].

2.3.7 Bayesian information criterion

Among the common existing model selection criteria, the Bayesian information criterion (BIC; Schwarz (1978)) and the Integrated completed likelihood (ICL; Biernacki et al. (2000)) constitute the reference choices in the recent literature on mixture models. The BIC is commonly used in model-based clustering and classifications applications involving a family of mixture models [Fraley & Raftery (2002); McNicholas & Murphy (2008)]. The use of BIC in the mixture model selection (Dasgupta &

Raftery (1998)), has been strongly encouraged. The BIC formula is given by

$$BIC = -2l(\hat{\boldsymbol{\Theta}}) + m \ln n. \tag{2.24}$$

Leroux (1992a) also shows that BIC does not underestimate the true number of component asymptotically. One potential problem of the BIC for model selection in model-based clustering applications is that a mixture component does not necessarily correspond to a true cluster. For example, a cluster might be represented by two mixture components.

2.3.8 Integrated completed likelihood

In an attempt to focus model selection on clusters rather than mixture components, Biernacki et al. (2000) proposed a penalized version of BIC called Integrated completed likelihood which is the BIC penalized for the estimated mean entropy;

$$ICL = BIC + \sum_{i=1}^{N}\sum_{g=1}^{G} MAP(\hat{z}_{ig}) \ln \hat{z}_{ig} \tag{2.25}$$

where $\sum_{i=1}^{N}\sum_{g=1}^{G} MAP(\hat{z}_{ig}) \ln \hat{z}_{ig}$ is the estimated mean entropy which reflects the uncertainty in the clustering of observation i into component G. Therefore, the ICL should be less likely compared to the BIC to split one cluster into two mixture components. Biernacki et al. (2000), based on numerical experiments, suggest to adopt the BIC and the ICL for indirect and direct applications, respectively.

2.3.9 Approximate Weight of Evidence

Banfield & Raftery (1993) suggest an approximate Bayesian solution to the choice of the number of components using classification ML approach. Their crude approximation to twice the log Bayes factor of G clusters leads to the approximate weight of evidence (AWE) criterion having the form

$$\text{AWE} = -2l(\hat{\boldsymbol{\Theta}}) + 2m(3/2 + \log n). \tag{2.26}$$

Definition and key reference for the adopted likelihood-based information criteria are given below

Table 2.2: Definition and key reference for the likelihood-based information criteria.

Information Criteria	Definition	Reference
AIC	$-2l(\hat{\boldsymbol{\Theta}}) + 2m$	Akaike (1973)
BIC	$-2l(\hat{\boldsymbol{\Theta}}) + m \log n$	Schwarz (1978)
AWE	$-2l(\hat{\boldsymbol{\Theta}}) + 2m(3/2 + \log n)$	Banfield & Raftery (1993)
AIC_3	$-2l(\hat{\boldsymbol{\Theta}}) + 3m$	Bozdogan (1994)
AIC_c	$\text{AIC} - 2\frac{m(m+1)}{n-m-1}$	Hurvich & Tsai (1989)
AIC_u	$\text{AIC}_c - n \log \frac{n}{n-m-1}$	McQuarrie et al. (1997)
ICL	$\text{BIC} + \sum_{i=1}^{n}\sum_{g=1}^{G} \text{MAP}(\hat{z}_{ig}) \ln \hat{z}_{ig}$	Biernacki et al. (2000)
CAIC	$-2l(\hat{\boldsymbol{\Theta}}) + m(1 + \log n)$	Bozdogan (1987)

2.3.10 Adjusted Rand Index

In addition to likelihood-based criteria presented in Table (2.2), Adjusted Rand Index (ARI) and its variants are also used for model selection. The data analyses are mainly conducted as clustering examples, the clustering results are compared with the *a priori* known true classifications. The original Rand Index [RI: Rand (1971)] is based on pairwise comparisons and is obtained by dividing the number of agreements (observations that truly agree and observations that truly disagree) by the total number of pairs. RI assumes values on $[0, 1]$, where 0 indicates no pairwise agreements between the MAP clustering and true cluster membership and 1 indicates perfect agreement. One criticism of RI is that its expected value is greater than 0, making smaller values difficult to interpret. ARI adjusts RI for chance by allowing for the possibility that classification performed randomly should correctly classify some observations. Thus, ARI has an expected value of 0 and perfect classification would result in a value of 1. Thus the higher the ARI, the better.

The component G that has the highest agreement with the true class label is selected as the best number of components. However, we observed that even the incorrect

number of component G can have the highest ARI, this could possibly mean that the class labels are not evenly distributed, since the ARI only counts the predicted class by the model that matches the true class, irrespective of their locations.

2.4 Variational Bayes Method

The development of variational techniques for Bayesian inference followed two parallel, yet separate tracks. Peterson & Anderson (1987) is arguably the first variational procedure for a particular model: a neural network. Their work, along with insights from statistical mechanics [Parisi (1988)], led to a flurry of VB procedures for a wide class of models [Saul et al. (1996); Jaakkola & Jordan (1996); Jaakkola & Jordan (1997); Ghahramani & Jordan (1997); Jordan et al. (1999)]. In parallel, Hinton & Camp (1993) proposed a variational algorithm for a similar neural network model. Neal & Hinton (1999) (first published in 1993) made important connections to the expectation maximization (EM) algorithm [Dempster et al. (1977)], which then led to a variety of variational Bayesian algorithms for other types of models [Waterhouse et al. (1996); MacKay (1997)].

2.4.1 History of Variational Methods

In the $18th$ century, Variational methods have their source from the work of Euler, Lagrange and others on the *calculus of variations* that is concerned with studying functionals. Standard calculus concerns the derivatives of functions. A function can be thought of as mapping the input value of a variable to the output value of the function. The derivative of the function then describes the various output values as we make infinitesimal changes to the input values. Similarly, a *functional* can be defined as a mapping that takes a function as an input and then returns the value of the functional as the output. For example, an entropy $H(p)$ takes a probability distribution $p(\boldsymbol{\theta})$ as an input and returns the quantity

$$H[p] = \int p(\boldsymbol{\theta}) \ln p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.27}$$

as the output. Additionally, *functional derivative* expresses how the value of the functional changes in response to the infinitesimal changes to the input function

[Feynman et al. (1964)]. The rules for the calculus of variations are discussed in [Appendix D]. Many problems can be expressed in terms of an optimization problem in which the optimized quantity is a functional. The solution is obtained by exploiting a search space for all possible input functions with the goal to find an optimal value that minimizes, or maximizes the objective function. Variational methods have wide applicability in areas such as maximum entropy [Schwarz (1988)] and finite element methods [Kapur (1989)]. In general, variational methods lend themselves to finding approximate solutions by restricting the range of optimized functions. For instance, consider a quadratic functions or a function with a linear combination of fixed basis functions in which only the coefficients of the linear combination can be random [Jordan et al. (1999); Jaakkola (2001)].

2.4.2 General formulation of Variational methods

Now, we consider in more detail the concept of variational optimization and how it can be applied to the inference problems. Suppose there is a fully Bayesian model where all parameters are treated as random variables, i.e. all parameters are given prior distributions. The model may also consist of both latent and parameters, the set of which in this case shall be denoted by $\mathcal{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$. Similarly, we shall denote the set of observed variables by $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where N denotes the set of the independent, identically distributed data. The main goal is to find an approximation for the posterior distribution $p(\mathcal{Z}|\mathcal{X})$ and the *evidence* $p(\mathcal{X})$ from a probabilistic model which specifies the joint distribution $p(\mathcal{X}, \mathcal{Z})$. According to the KL divergence discussed above, we decompose KL divergence to obtain the log marginal probability to maximize

$$-\mathrm{KL}(q||p) = \int q(\mathcal{Z}) \log \frac{p(\mathcal{Z}|\mathcal{X})}{q(\mathcal{Z})} d\mathcal{Z} \tag{2.28}$$

$$= \int q(\mathcal{Z}) \Big[\log p(\mathcal{Z}|\mathcal{X}) - \log q(\mathcal{Z}) \Big] d\mathcal{Z} \tag{2.29}$$

by Bayes' rule we have

$$= \int q(\mathcal{Z}) \Big[\log \frac{p(\mathcal{X}, \mathcal{Z})}{p(\mathcal{X})} - \log q(\mathcal{Z}) \Big] d\mathcal{Z} \tag{2.30}$$

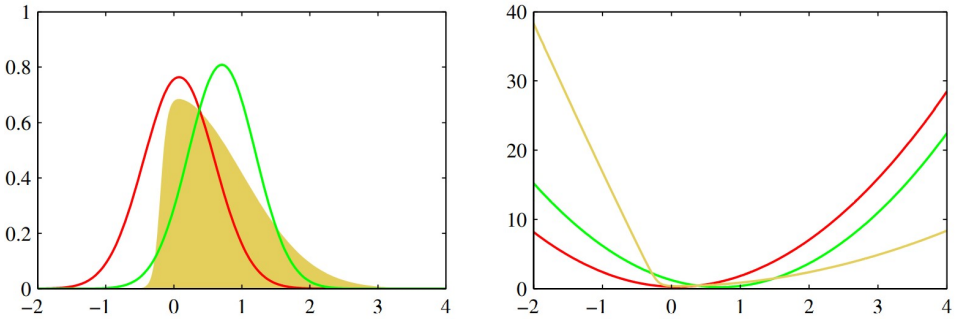


Figure 2.1: Illustration of the variational (green) and Laplace (red) approximations compared to the original distribution (yellow), and the right-hand plot shows the negative logarithms of the corresponding curves. Source: [Bishop (2006)]

$$= \int q(\mathcal{Z}) \log \frac{p(\mathcal{X}, \mathcal{Z})}{q(\mathcal{Z})} d\mathcal{Z} - \int q(\mathcal{Z}) \log p(\mathcal{X}) d\mathcal{Z} \tag{2.31}$$

Since $\log p(\mathcal{X})$ has no variable $\mathcal{Z}$ then the right-hand side of Equation (2.31) is constant. Then,

$$-\mathrm{KL}(q||p) = \mathcal{L}(q) - \log p(\mathcal{X}). \tag{2.32}$$

Then, the marginal distribution is

$$\log p(\mathcal{X}) = \mathrm{KL}(q||p) + \mathcal{L}(q). \tag{2.33}$$

where

$$\mathcal{L}(q) = \int q(\mathcal{Z}) \log \frac{p(\mathcal{X}, \mathcal{Z})}{q(\mathcal{Z})} d\mathcal{Z}. \tag{2.34}$$

The variational method can be seen as a generalized EM algorithm in that the parameter vector $\boldsymbol{\theta}$ is no longer fixed but random or stochastic variables and are embedded in $\mathcal{Z}$. We note that we have assumed continuous variables and the same settings are applicable for discrete variables with summation rather than integration. We then maximize the lower bound given in Equation (2.34) by optimizing w.r.t the approximate distribution $q(\mathcal{Z})$, which is equivalent to minimizing the KL divergence. The maximum of the lower bound occurs when the KL divergence vanishes, i.e. when approximate distribution equals the posterior distribution $q(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X})$. We therefore work with a restricted family of distributions $q(\mathcal{Z})$ and then seek the member of this family that minimizes the KL divergence. The main goal is that the restricted

family sufficiently comprises of only tractable distributions, while the family is rich and flexible enough to provide a good approximation to the true posterior distribution. It is important to emphasize that the restriction is to achieve tractability, and that this requirement should prompt a rich family of approximating distribution as possible because in particular, there is no 'over-fitting' attached with highly flexible distributions. However, using highly flexible approximating distributions helps us to approach the true posterior distribution more intimately.

One way of restricting the family of approximating distributions is to use a parametric distribution $q(\mathcal{Z}|\mathbf{\Theta})$ parameterized by $\mathbf{\Theta}$. The lower bound $\mathcal{L}(q)$ becomes a function of $\mathbf{\Theta}$, and we exploit standard nonlinear optimization techniques to determine the optimal values for the parameters. The explanation of this framework is given in Figure (2.1), in which the variational distribution is a Gaussian optimized with respect to its mean and variance.

2.4.3 The mean-field variational family

We hereby consider another way to restrict the family of distributions $q(\mathcal{Z})$. Suppose the elements of $\mathcal{Z}$ is partitioned into a disjoint group denoted by $\mathcal{Z}_j$, where $j = 1, ..., J$. We then say that the approximate distribution is factorized with respect to the group, so that

$$q(\mathcal{Z}) = \prod_{j=1}^{J} q_j(\mathcal{Z}_j) \tag{2.35}$$

This factorized form of variational inference corresponds to a framework developed in physics called *mean field theory* [Parisi (1988)]. We now seek that distribution with which the lower bound $\mathcal{L}(q)$ is maximized among all the distributions $q(\mathcal{Z})$. The goal is to make a free form variational optimization of $\mathcal{L}(q)$ with respect to all the distributions $q_j(\mathcal{Z}_j)$. This is optimized with respect to each of the factors in a coordinate-wise or in parallel by substituting Equation (2.35) in Equation (2.34). We then marginalize out the dependence on one of the factors $q_l(\mathcal{Z}_l)$ where $q_l(\mathcal{Z}_l)$ denotes the distribution that does not depend on the factor q_j. We therefore obtain the following

$$\mathcal{L}(q) = \int \prod_j q_j \Big[\log \frac{p(\mathcal{X}, \mathcal{Z})}{\prod_j q_j} \Big] \, d\mathcal{Z} \tag{2.36}$$

$$= \int \prod_j q_j \Big[\log p(\mathcal{X}, \mathcal{Z}) - \sum_j \log q_j \Big] \, d\mathcal{Z} \tag{2.37}$$

$$= \int q_l \Big[\int \log p(\mathcal{X}, \mathcal{Z}) \prod_{j \neq l} q_j \, d\mathcal{Z}_j \Big] d\mathcal{Z}_l - \int q_l \log q_l \, d\mathcal{Z}_l + \text{const} \tag{2.38}$$

$$= \int q_l \Big\{ \mathrm{E}_{j \neq l} \Big[\log p(\mathcal{X}, \mathcal{Z}) \Big] + \text{const} \Big\} \, d\mathcal{Z}_l - \int q_l \log q_l \, d\mathcal{Z}_l + \text{const} \tag{2.39}$$

Here, the notation $\mathrm{E}_{j \neq l}[...]$ denotes an expression with respect to the q distributions over all variables $\mathbf{z}_j$ for $j \neq l$.

Suppose we maximize the lower bound $\mathcal{L}(q)$ in Equation (2.39) w.r.t. all possible forms for the distribution $q_l(\mathcal{Z}_l)$ and keep all other variables $q_{j \neq l}$ fixed. Thus maximizing Equation (2.39) is equivalent to minimizing the Kullback-Leibler divergence because Equation (2.39) is a negative KL divergence. Thus we obtain a general expression for the optimal solution $q_l^*(\mathcal{Z}_l)$ given by

$$\log q_l^*(\mathcal{Z}_l) = \mathrm{E}_{j \neq l}[\log p(\mathcal{X}, \mathcal{Z})] + \text{const.} \tag{2.40}$$

Equation (2.40) says that the log of the optimal solution for factor q_l is obtained by taking the log of the joint distribution over all latent and observed variables and then averaging by taking the expectation with respect to all other factors q_j independent of q_l for $j \neq l$. Taking the exponential and normalize Equation (2.40) gives the following

$$q_l^*(\mathcal{Z}_l) = \frac{\exp(\mathrm{E}_{j \neq l}[\log p(\mathcal{X}, \mathcal{Z})])}{\int \exp(\mathrm{E}_{j \neq l}[\log p(\mathcal{X}, \mathcal{Z})]) \, d\mathcal{Z}_l} \tag{2.41}$$

The set of equations given in Equation (2.40) for $l = 1, ..., J$ represents a set of consistency conditions for the maximum of the lower bound subject to the factorization constraint. We will first seek a consistent solution by first initializing all of the factors $q_j(\mathcal{Z}_l)$ appropriately and then cycling through the factors and replacing each in parallel with a revised estimate given in the Equation (2.40) evaluated using the current estimates for all other factors. Convergence is discussed in [Boyd & Vandenberghe (2004)]. This is guarantee due to a convex bound with respect to each of the factors $q_j(\mathcal{Z}_j)$.

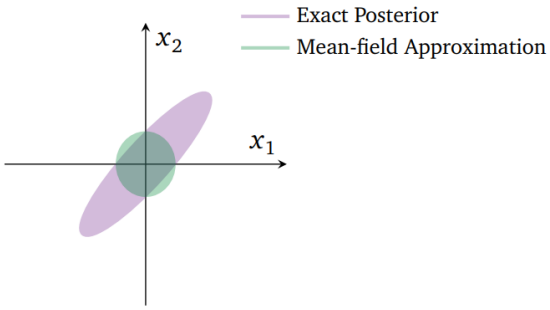


Figure 2.2: Visualizing the mean-field approximation to a two-dimensional Gaussian posterior. The ellipse show the effect of mean-field factorization. Source: [Blei et al. (2018)]

2.4.4 When should a statistician use VB or MCMC?

A critical question of interest is when should a statistician prefer one approximate method to another. For instance, which method should be preferred between Variational Bayes (VB) and MCMC. Some researchers have delved deeper into this question and some have unified the two algorithms into one algorithm. Considering the difference between VB and MCMC will enlighten us on which one to prefer over the other.

MCMC methods are computationally intensive than VB but produce somewhat asymptotically exact samples from a target density than VB [Robert & Casella (2004)]. VB on the other hand, lacks such an accolade of producing exact posterior distribution, but only finds a density close to the target density. However, it tends to be faster than MCMC. VB utilizes optimization and takes advantage of methods like stochastic optimization [Robbins & Monro (1951); Kushner & Yin (1997)]. In the field of large data, VB is suitable than MCMC and in the scenario where many models are to be explored. In contrast, MCMC is suitable for smaller data and in the situation where we are happy to pay the computational cost for accuracy. For example, we might use MCMC where we require precise inferences. But we might use VB when fitting a probabilistic model to billions of text documents and where the inferences will be used to serve as a search results to a large population of users.

Another factor that determines which approximate method to prefer is the geometry of the posterior distribution. For example, a posterior distribution in a mixture

models admits multiple modes. According to the model, Gibbs sampling tends to be a powerful approach to sampling from the target distributions as it quickly focuses on one of the modes. However, when Gibbs is not a better choice, VB can perform better than a general MCMC method e.g. Hamiltonian Monte Carlo e.g. Hamiltonian Monte Carlo for smaller datasets [Kucukelbir et al. (2015)].

One limitation of VB is that it underestimates the variance of the posterior density. This is visualized in Figure (2.2). This depending on the objective, underestimating the variance may be acceptable. Several areas of research have shown that VB does not necessarily suffer in accuracy, especially in terms of posterior predictive densities [Blei & Jordan (2006); Braun & McAuliffe (2010); Kucukelbir et al. (2016)].

2.5 Expectation Propagation Methods

2.5.1 General EP

An alternative form of deterministic approximate inference which uses an inclusive type of KL divergence is known as *expectation propagation* or EP for short [Minka (2001)]. Similar to variational Bayesian, EP also minimizes Kullback-Leibler divergence but in an inclusive or reverse form, which gives the approximation rather different properties. We consider the problem of minimizing $\mathrm{KL}(p||q)$ with respect to the approximate distribution $q(\boldsymbol{\theta})$ when the $p(\boldsymbol{\theta})$ is a fixed distribution and $q(\boldsymbol{\theta})$ is a member of the exponential family which can be written as

$$q(\boldsymbol{\theta}) = h(\boldsymbol{\theta})g(\boldsymbol{\zeta})\exp(\boldsymbol{\zeta}^T\boldsymbol{u}(\boldsymbol{\theta})). \tag{2.42}$$

As a function of parameter $\boldsymbol{\zeta}$, Kullback-Leibler divergence becomes

$$\mathrm{KL}[p||q] = \int p(\boldsymbol{\theta})\log p(\boldsymbol{\theta})d\boldsymbol{\theta} - \int p(\boldsymbol{\theta})\log[h(\boldsymbol{\theta})g(\boldsymbol{\zeta})\exp(\boldsymbol{\zeta}^T\boldsymbol{u}(\boldsymbol{\theta}))], \tag{2.43}$$

embedding all quantity that is independent of parameter $\boldsymbol{\zeta}$, we have

$$= -\log g(\boldsymbol{\zeta}) - \boldsymbol{\zeta}^T\mathbb{E}_{p(\boldsymbol{\theta})}\Big[\boldsymbol{u}(\boldsymbol{\theta})\Big] + \text{const.} \tag{2.44}$$

We can minimize KL$[p||q]$ by setting the gradient with respect to $\boldsymbol{\zeta}$ to zero and this gives the following

$$-\nabla \log g(\boldsymbol{\zeta}) = \mathbb{E}_{p(\boldsymbol{\theta})}\Big[\boldsymbol{u}(\boldsymbol{\theta})\Big] \tag{2.45}$$

By using the method of *moment matching*, we equate the expectation of $\boldsymbol{u}(\boldsymbol{\theta})$ under the distribution $q(\boldsymbol{\theta})$ to the expectation in equation (2.45) as follows;

$$\mathbb{E}_{q(\boldsymbol{\theta})}[\boldsymbol{u}(\boldsymbol{\theta})] = \mathbb{E}_{p(\boldsymbol{\theta})}[\boldsymbol{u}(\boldsymbol{\theta})] \tag{2.46}$$

The optimal solutions simply corresponds to matching the expectation sufficient statistics. For example, if $q(\boldsymbol{\theta})$ is a Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ then the Kullback-Leibler divergence KL$[p||q]$ is minimized by setting the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of $q(\boldsymbol{\theta})$ to the mean and covariance of $p(\boldsymbol{\theta})$.

2.5.2 Practical Algorithm for EP

Now, we exploit this result to obtain a practical algorithm for approximate inference. For many probabilistic models, the joint distribution of data $\mathcal{D}$ and latent variables including the parameters $\boldsymbol{\theta}$ can be seen as a product of factors as follows;

$$p(\boldsymbol{\theta}, \mathcal{D}) = \prod_i t_i(\boldsymbol{\theta}). \tag{2.47}$$

This arises for example in independent, identically distributed data in which there is one factor $t_i(\boldsymbol{\theta}) = p(\mathbf{x}_i|\boldsymbol{\theta})$ for each data point $\mathbf{x}_i$, where $i = 0, ..., N$ with a factor $t_0 = p(\boldsymbol{\theta})$ corresponding to the prior distribution. We are generally interested in evaluating the intractable posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ for prediction making and computing the model evidence $p(\mathcal{D})$ for the purpose of model comparison. The posterior is given by

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})}\prod_i t_i(\boldsymbol{\theta}) \tag{2.48}$$

and the model evidence is given by

$$p(\mathcal{D}) = \int \prod_i t_i(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.49}$$

The setting remains the same for discrete variable with integral replaced by summations. We hereby suppose that the marginalization over $\boldsymbol{\theta}$ and the marginalization with respect to the posterior distribution required to make predictions are intractable so the only solution is approximation.

Expectation propagation is hinged on an approximation posterior distribution which is also factorized as follows

$$q(\boldsymbol{\theta}) = \frac{1}{Z}\prod_i \tilde{t}_i(\boldsymbol{\theta}) \tag{2.50}$$

where each factor $\tilde{t}_i(\boldsymbol{\theta})$ in the approximation corresponds to one of the factors $t_i(\boldsymbol{\theta})$ in the true posterior distribution in Equation (2.48) and the factor $1/Z$ is the normalizing constant needed to make Equation (2.50) a PDF, i.e. integrate to unity. We shall assume that the approximate factor $\tilde{t}_i(\boldsymbol{\theta})$ comes from the exponential family and the product of the factors will therefore be from the exponential family. For example, if each of the $\tilde{t}_i(\boldsymbol{\theta})$ is taken to be a Gaussian, the overall approximation $q(\boldsymbol{\theta})$ will also be a Gaussian. Generally, we would like to minimize the Kullback-Leibler divergence between the true posterior and the approximate posterior

$$\mathrm{KL}[p||q] = \mathrm{KL}\left(\frac{1}{p(\mathcal{D})}\prod_i t_i(\boldsymbol{\theta})\middle\|\frac{1}{Z}\prod_i \tilde{t}_i(\boldsymbol{\theta})\right) \tag{2.51}$$

Equation (2.51) is a reverse form of the Kullback-Leibler divergence used in variational Bayesian. However, the involvement of the averaging with respect to the true posterior distribution makes the Equation (2.51) intractable. To obtain a tractable version of Equation (2.51), we instead minimize the KL divergences between the corresponding pairs $t_i(\boldsymbol{\theta})$ and $\tilde{t}_i(\boldsymbol{\theta})$ of factors. However, the product of the factors could produce a poor approximation because each factor is individually approximated.

Expectation propagation provides a much better approximation by optimizing each factor in turn in the context of all of the remaining factors. It starts by initializing the approximate factors $\tilde{t}_i(\boldsymbol{\theta})$, and cycles the factor refining them one at a time. This is similar in the same spirit with the update of factors in the variational Bayes framework discussed earlier. Suppose we wish to refine factor $\tilde{t}_i(\boldsymbol{\theta})$, the refined factor is first removed from the approximate posterior $q(\boldsymbol{\theta})$ to obtain the *cavity distribution*

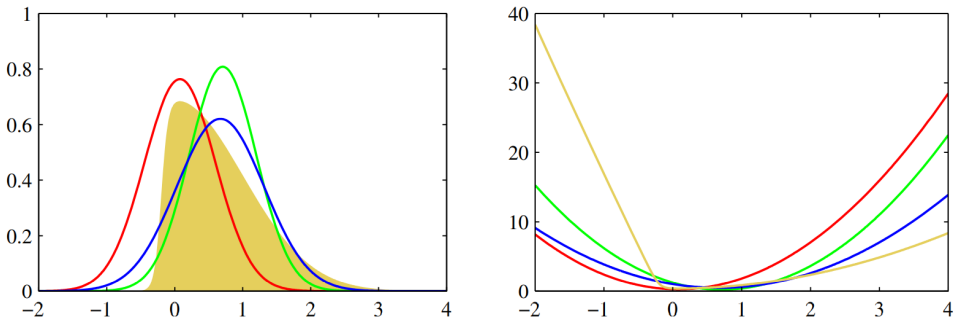


Figure 2.3: Illustration of the Expectation Propagation (blue), variational Bayes (green) and Laplace (red) approximations compared to the original distribution (yellow), and the right-hand plot shows the negative logarithms of the corresponding curves. Source: [Bishop (2006)]

$q_{-i}(\boldsymbol{\theta})$.

$$q_{-i}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\tilde{t}_i(\boldsymbol{\theta})}. \tag{2.52}$$

Equation (2.52) is now combined with the factor $t_i(\boldsymbol{\theta})$ to compute the *tilted posterior*

$$\hat{p}_i(\boldsymbol{\theta}) = \frac{1}{Z_i} t_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta}) \tag{2.53}$$

where Z_i is the normalization constant given by

$$Z_i = \int t_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{2.54}$$

Conceptually, we now determine the revised form of the approximate term $\tilde{t}_i(\boldsymbol{\theta})$ by ensuring that the product and minimizing the Kullback-Leibler divergence KL$[\hat{p}||q]$

$$q^{new}(\boldsymbol{\theta}) \propto \tilde{t}_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta}) \tag{2.55}$$

is as close as possible to

$$\hat{p}_i(\boldsymbol{\theta}) \propto t_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta}) \tag{2.56}$$

through minimizing the Kullback-Leibler divergence. This is easily solved when the approximate posterior $q^{new}(\boldsymbol{\theta})$ is from an exponential family, so we can find the best approximate posterior close to the tilted posterior by matching the parameters of $q^{new}(\boldsymbol{\theta})$ with that of the tilted posterior. Finally, we compute the update for the

approximate term $\tilde{t}_i(\boldsymbol{\theta})$.

$$\tilde{t}_i(\boldsymbol{\theta}) = \tilde{Z}_i \frac{q^{new}(\boldsymbol{\theta})}{q_{-i}(\boldsymbol{\theta})} \tag{2.57}$$

where $\tilde{Z}_i$ is found by matching the zeroth-order moments as follows

$$\tilde{Z}_i = \int \tilde{t}_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta})\ d\boldsymbol{\theta} = \int t_i(\boldsymbol{\theta})\ q_{-i}(\boldsymbol{\theta})\ d\boldsymbol{\theta} = Z_i \tag{2.58}$$

Practically, several passes are made through the set of factors, updating each in turn. The posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ is then approximated using the approximate posterior and the evidence $p(\mathcal{D})$ using the

$$p(\mathcal{D}) \approx \int \prod_i \tilde{t}_i(\boldsymbol{\theta})\ d\boldsymbol{\theta} \tag{2.59}$$

A special case of EP si known as *assumed density filtering* (ADF) [Maybeck (1982); Lauritzen (1992); Boyen & Koller (1998); Opper & Winther (1999)]. It is obtained by first initializing the first approximating factor in EP to unity. The remaining factors are initialized and then make one pass through the factors by updating each of them once. ADF is found to be appropriate for an on-line learning in which data points are learned sequentially. Each data point is learned and then discarded before considering the next point. However, EP technique is suitable for a batch setting where we have the opportunity to re-use the data points many times in order to achieve improved accuracy. One major limitation in ADF is the sequential nature which has a dependence on the order in which the data points are updated. This limitation is overcome by EP [Miele, Cragg & Levy (1971)].

2.5.3 Algorithm for General EP

Algorithm 2 The General EP Algorithm

Given a probability model, we have a joint distribution over a set of observed data $\mathcal{D}$, and stochastic variables $\boldsymbol{\theta}$.

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i t_i(\boldsymbol{\theta})$$

we wish to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ by a distribution of the form

$$q(\boldsymbol{\theta}) = \frac{1}{Z}\prod_i \tilde{t}_i(\boldsymbol{\theta})$$

We now approximate the model evidence $p(\mathcal{D})$ as follows

1. Initialize all of the approximating factors $\tilde{f}_i(\boldsymbol{\theta})$.
2. Initialize the posterior approximation by setting
$$q(\boldsymbol{\theta}) \propto \prod_i \tilde{t}_i(\boldsymbol{\theta}).$$
3. Until convergence:
   (a) Choose a factor $\tilde{t}_i(\boldsymbol{\theta})$ to refine.
   (b) Remove $\tilde{t}_i(\boldsymbol{\theta})$ from the posterior by division
   $$q_{-i}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\tilde{t}_i(\boldsymbol{\theta})}.$$
   (c) Compute the new approximate posterior q^{new} by setting its moments equal to tilted posterior $\hat{p}_i(\boldsymbol{\theta})$, with the normalization constant
   $$Z_i(\boldsymbol{\theta}) = \int q_{-i}(\boldsymbol{\theta}) t_i(\boldsymbol{\theta})$$
4. Evaluate the approximation to the model evidence
$$p(\mathcal{D}) \approx \int \prod_i \tilde{t}_i(\boldsymbol{\theta})\, d\boldsymbol{\theta}.$$

2.5.4 Comparison VB and EP

The lack of guarantee of no convergence in the EP iterations makes it different from VB. VB iteratively maximizes a lower bound on the log marginal likelihood, in which each iteration is guaranteed to increase the bound. It is possible to optimize the cost function of EP directly which provide a guarantee of convergence. However, the resulting algorithms can be slower or more complex to implement.

Another difference between EP and VB stems from the types of Kullback-Leibler divergence they both minimize. EP minimizes $\mathrm{KL}[p||q]$ while VB minimizes $\mathrm{KL}[q||p]$. As we have discussed above, EP competes well with VB in terms of accuracy. A simulation study shows that expectation propagation is somewhat more accurate than mean field variational Bayes for larger sample sizes, but at the cost of considerably more algebraic and computational effort [Kim & Wand (2016)].

Chapter 3

Variational Bayesian: EM – OLS Cluster Weighted Models

3.1 Background History

Finite mixture of distributions has gained a long standing in statistical modeling due to its mathematical-based approach to a wide random phenomena. Because of their extremely flexible method of modeling, finite mixture models have brought to limelight and received increasing attention over the years from both practical and theoretical viewpoints. Indeed, in the past decades the extent and the potentials of the applications of finite mixture models have widened considerably. Finite mixture models have been widely and successfully applied in many fields such as biological, genetics, medicine, psychiatry, economics, engineering, marketing, astronomy, among many other fields in the biological, physical, and social sciences. In these applications, finite mixture models underpin a variety of techniques in major areas of statistics, including latent class analysis, discriminant analysis, image analysis, and survival analysis, in addition to their more direct role in data analysis and inference of providing descriptive models for distributions.

The usefulness of mixture distributions in the modeling of heterogeneity in a cluster analysis context is obvious. In case where there is a group structure, they have a very useful role in assessing the error rate such as sensitivity and specificity of diagnostic and screening procedures in the absence of a gold standard. But as any continuous

distribution can be approximated arbitrarily well by finite mixture of normal densities with common variance or covariance matrix in case of multivariate, mixture models provide a convenient semi-parametric or nonfunctional framework in which to model unknown distributional shapes, whatever the objective or the estimation or the flexible construction of Bayesian priors. One of the examples that underpinned the evolution of finite mixture models is the demonstration that with $N = 10,000$ observation, a log normal densities can be successfully approximated by mixture of about 30 normal distributions [Priebe (1994)]. In contrast, a kernel density estimator uses a mixture of $10,000$ normal distributions. A mixture model is able to model quite complex distributions through an appropriate choice of its components to represent accurately the local areas of support of the true distribution. It can also handle situations where a single parametric family will fail to provide a satisfactory model for local variations in the observed data. Inferences about the modeled phenomenon can be made without difficulties from the mixture components, since the latter are chosen for their tractability.

3.1.1 Former Approach to Mixture Analysis

One of the first major analysis involving the use of mixture models can be dated back to 100 years ago by the famous biometrician [Pearson (1894)], who in his classic paper fitted a mixture of two normal density functions with different means μ_1 and μ_2 and variances σ_1^2 and σ_2^2 in proportions π_1 and π_2 to some data provided by [Weldon (1892); Weldon (1893)].

The data set analyzed consisted of measurements on the ratio of forehead to body length of $N = 1000$ crabs sampled from the Bay of Naples. These measurements recorded in the form of $v = 29$ intervals are displayed in McLachlan & Peel (2000). Weldon (1893) speculated that the skewness in the histogram of these data might be a signal that this population was evolving from two new subspecies.

Pearson (1894) mixture model-based approach suggested that there were presence of heterogeneity. To estimate the models, Person used the method of moments to fit this mixture model to the mid-points of the intervals. However, McLachlan & Peel (2000) used method of maximum of likelihood to fit the same model, which gives a

fit very similar to that obtained by Pearson's. Indeed, as Everitt (1996) noted about the computational effort in fitting this model which must have been at the time a daunting prospect to the users of the mixture models, this daunting prospect led to Charlier (1996) statement "*The solution of an equation of the ninth degree, where almost all power to the ninth of the unknown quantity are existing*". He confirmed the effort Pearson possessed in performing this heroic task yet feared if he would have successors, if the dissection of the frequency curve into two components is not very urgent. Not surprisingly, various attempts have been made over the years to simplify Pearson's moments-based approach to fitting of a normal mixture model.

3.1.2 Impact of EM Algorithm

It has been about 20 years that considerable advances have been made in the fitting of finite mixture models, in particular by the method of maximum likelihood. There had been some reluctance in the past to fitting mixture models to data even with the advent of high-speed due to the lack of understanding of issues arising with their fitting such as multiple maxima and unboundedness of likelihood function in case of normal components with unequal covariance matrices. However, with the clarity and proper understanding of these computational issues, it has led to the increasing use of mixture models in practice. Fitting of finite mixture models by maximum likelihood had been studied in the literature such as [Day (1969); Wolfe (1965); Wolfe (1967); and Wolfe (1970)]. However, the study of EM algorithm by Dempster et al. (1977) greatly stimulated interest in the use of finite mixture distributions to model different subpopulation in the observed data. The reason is that fitting of mixture models by maximum likelihood is a classical example of a problem that is considerably simplified by the EM's conceptual unification of Maximum Likelihood Estimation (MLE) from the data that can be viewed as being incomplete. This was confirmed by [Aitkin & Aitkin (1994)] that the application of mixture modeling reported in the literature increased in number after Dempster's study of EM algorithm. This also applies to the applications in this thesis.

3.2 Basic Definition of FMM

Let $\boldsymbol{X}_1, ..., \boldsymbol{X}_N$ denote a random sample of size N, where $\boldsymbol{X}_i$ is a d-dimensional random vector with Probability Density Function (PDF) $p(\boldsymbol{x}_i)$ on $\mathbb{R}^d$. In practice, $\boldsymbol{X}_i$ contains the random variables corresponding to d measurements on the ith individual of some features on the phenomenon under study. Let $\boldsymbol{X} = (\boldsymbol{X}_i', .., \boldsymbol{X}_N')'$ denote the vector form of $\boldsymbol{X}$ accomplished by transpose and the realization of the random vector is denoted by the corresponding lower-case letter, i.e., $\boldsymbol{x} = (\boldsymbol{x}_i', ..., \boldsymbol{x}_N')'$ denotes an observed random sample where $\boldsymbol{x}_i$ is the observed value of the random vector $\boldsymbol{X}_i$.

In the premise of continuous random vector $\boldsymbol{X}_i$, the density $p(\boldsymbol{x}_i)$ of $\boldsymbol{X}_i$ is written as follows;

$$p(\boldsymbol{x}_i) = \sum_{g=1}^{G} \pi_g p_g(\boldsymbol{x}_i), \tag{3.1}$$

where the $p_g(\boldsymbol{x}_i)$ are densities and the π_g are the nonegative quantities with constraints such as

$$0 \leq \pi_g \leq 1, \qquad g = 1, ..., G \tag{3.2}$$

and

$$\sum_{g=1}^{G} \pi_g = 1. \tag{3.3}$$

The quantities $\pi_1, ..., \pi_G$ are called the mixing probabilities, proportions or weights, and the $p_g(\boldsymbol{x}_i)$ are called component densities of the mixture. In this formulation, the number of components G can either be fixed a priori or inferred from the data, along with the mixing proportions and the parameters in the specified forms for the components' densities.

An obvious way of generating a random vector $\boldsymbol{X}_i$ with the G-component mixture density $p(\boldsymbol{x}_i)$, given in Equation (3.1), is as follows: Let $\boldsymbol{Z}_i$ be a categorical random variable taking on the values $1, ..., G$ with probabilities $\pi_1, ..., \pi_G$, respectively, and suppose that the conditional density of $\boldsymbol{X}_i | \boldsymbol{Z}_i = g$ is $p_g(\boldsymbol{x}_i)$. The variable $\boldsymbol{Z}_i$ is the component label of the feature vector $\boldsymbol{X}_i$ and the $p(\boldsymbol{x}_i)$ is the marginal density of $\boldsymbol{x}_i$. $\boldsymbol{Z}_i$ is distributed according to a multinomial distribution consisting of one draw on G

categories with probabilities $\pi_1, ..., \pi_G$; that is,

$$p(\boldsymbol{Z}_i = \boldsymbol{z}_i) = \prod_{g=1}^{G} \pi_g^{z_{jg}} \tag{3.4}$$

where $\boldsymbol{\pi} = (\pi_1, ..., \pi_G)'$. It is convenient to work with a G-dimensional component-label vector $\mathbf{Z}_i$ in place of the single categorical variable Z_i, where the gth element of $\mathbf{Z}_i$, $Z_{ig} = (\mathbf{Z}_i)_g$, is either one or zero according to whether the component of the origin of $\boldsymbol{X}_i$ in the mixture is equal to g or not for $g = 1, ..., G$.

The direct application of the above interpretation is where $\boldsymbol{X}_i$ is drawn from a population $\mathcal{D}$ which can be decomposed to $\mathcal{D}_1, ..., \mathcal{D}_G$ with proportions $\pi_1, ..., \pi_G$. For example, a source of the heterogeneity is often age, sex, species, geographical origin, and cohort status. Also, a population $\mathcal{D}$ may consist of two groups $\mathcal{D}_1$ and $\mathcal{D}_2$, corresponding to those members with or without disease under study. Some components may be obvious and known a priori from the external grouping while some may not be identified with the externally existing groups. The essence of introducing the component into the mixture model is to allow for greater flexibility in modeling a heterogeneous population that is apparently impossible with a single component distribution.

Mixture models can be seen as an approach that hinges between nonparametric and parametric approaches to statistical estimation. Mixture model-based approaches are parametric in that parametric forms $p_g(\boldsymbol{x}_i)$ are specified for the component density functions, but also take a nonparametric form in that they can be allowed the number of components G can be allowed to grow.

3.2.1 Advent of EM Algorithm for Mixture Models

With the advent of high-speed computers, attention was shifted to ML estimation of the parameters in a mixture distribution. Rao (1948) first used Fisher's method of scoring for a mixture of two univariate distributions with equal variances. However, Butler (1986) argued that Newcomb (1886) predating Pearson (1894) attempt suggested an iterative reweighting scheme which was similar to EM algorithm of Dempster et al. (1977) to compute the MLE of the common mean of a mixture in a known

proportions of a finite number of univariate normal populations with known variances. Also, Jeffrey (1932) used essentially the EM algorithm in iteratively computing the estimates of the means of two univariate normal populations which had known variances and which were mixed in known proportions. Until Hasselblad (1966), Hasselblad (1969) addressed the problem of EM, MLE did not experience a resurgence. Initially, he applied it to a mixture of g univariate normal distributions with equal variances, and then to mixtures of distribution from the exponential family. He also presented the solutions in an iterative form which corresponded to particular applications of the EM algorithm of Dempster et al. (1977). There were many other works on ML estimation of mixture with the computation of the estimates expressed in this iterative form such as Duda & Hart (1973), Hosmer (1973*b*), Hosmer (1973*a*) and Peters & Coberly (1976).

However, Dempster et al. (1977) established the convergence properties of the ML solution for the mixture problem on a theoretical basis by formalizing this iterative scheme in a general context through their EM algorithm. The evolution of EM had a positive impact on finite mixture models. There were quite an extensive literature on finite mixture models such as Everitt and Hand (1981), Titterington et al. (1985), McLachlan & Basford (1988) and so on.

Continuing from Equation (3.1), finite mixture model was presented in a parametric form after the resurgence of EM algorithm for the MLE. Equation (3.1) can be written as

$$p(\boldsymbol{x}_i; \boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g p_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g) \tag{3.5}$$

where the vector $\boldsymbol{\Theta}$ containing all the unknown parameters in the mixture model is written as

$$\boldsymbol{\Theta} = (\pi_1, ..., \pi_{G-1}, \boldsymbol{\delta}')' \tag{3.6}$$

and $\boldsymbol{\delta}$ is the vector containing all the parameters in $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_G$ known *a priori* be distinct. $\mathcal{D}$ denotes the specified parameter space for the $\boldsymbol{\Theta}$, and $\boldsymbol{\pi} = (\pi_1, ..., \pi_G)'$ is the vector of mixing weights. In defining $\boldsymbol{\Theta}$, π_G is simply omitted due to sum-up to unity.

3.2.2 The Evolution of Cluster Weighted Models

Most of the finite mixture models assume the *assignment independence* which implies that the probability for a point to be generated by one of the cluster must be the same for all the covariate values $\mathbf{x}$. On the other hand the assignment of data point into the cluster must be independent of the covariates [Hennig (2000)]. The cluster membership is determined by the covariate values. There are two reasonable models for linear regression clusters that do not assume assignment independence. One strategy used is to replace the fixed covariates by covariate distributions that are allowed to differ between the clusters. This is also similar to the evolution of cluster weighted models. It assumes the varying covariates with a parameterized family of distributions. This solves the problem of *assignment independence* i.e. the covariate distributions of the mixture components is unique across the cluster. In the framework of mixture models with varying covariates, the cluster weighted model [CWM; Gershenfeld (1997)], is given by the equation

$$p(y, \boldsymbol{x}) = \sum_{g=1}^{G} \pi_g p(y, \boldsymbol{x}|\mathcal{D}_g) = \sum_{g=1}^{G} \pi_g p(y|\boldsymbol{x}, \mathcal{D}_g) p(\boldsymbol{x}|\mathcal{D}_g), \tag{3.7}$$

also called the saturated mixture regression model [Wedel (2002)], constitutes a reference approach to model the joint density. In Equation (3.7), normality of both $p(y|\boldsymbol{x}, \mathcal{D}_g)$ and $p(\boldsymbol{x}|\mathcal{D}_g)$ is commonly assumed [Gershenfeld (1997); (n.d.)].

3.3 CWMs-tSNE for High-dimensional data

Given a data set of N pairs of points

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\} \tag{3.8}$$

with vector inputs $\boldsymbol{x}_i \in \mathbb{R}^d$ and corresponding scalar outputs $y_i \in \mathbb{R}$, the aim of modeling is to find an estimate of $\hat{y}$ of the output for a new vector observation $\boldsymbol{x}_{new} \notin \mathcal{D}$, which is often called a test data.
Many different approaches arise both in machine learning and statistics where one tries to find a good approximation for the *regression* $\mathrm{E}[Y|\mathbf{X}]$. The pairs of observation

$(\boldsymbol{x}_i, y_i)$ are the realizations of the random variables $\boldsymbol{X}, Y$, where the random variables are drawn from an unknown joint probability P. The regression $\mathrm{E}[Y|\boldsymbol{X}]$ is seen as random variable with a conditional expectation $f(\boldsymbol{x}) \equiv \mathrm{E}[Y|\boldsymbol{X} = \boldsymbol{x}]$. Customary to statistics, Y and $\boldsymbol{X}$ are called *dependent* and *independent* variables respectively. On the contrary, in machine learning, they are termed as *label* and *features*. In least square approach, $f(\boldsymbol{x}) \sim y_i$ is the best approximation for the output value y_i. The prediction capability of a learning method relates strongly to the generalization performance and is measured on an independent test data. In practice, the assessment is very important, since it guides the choice of learning model and gives us a measure of the quality of chosen model.

In this chapter, we will focus mainly on the application of CWMs to high-dimensional data. The data considered in this chapter ranges from the tens to hundreds of features. Like any machine learning classifiers, CWMs clustering performance can be hindered by the redundancies in the feature space of the data. Moreover, the computation speed reduces exponentially with increase in dimensionality. We hereby present CWMs in the face of high-dimensional data. We begin the discussion with an interplay between t-distributed stochastic neighbor embedding and CWMs for clustering high-dimensional data.

3.3.1 Cluster Weighted Models

Cluster Weighted Models denoted by CWMs hinges between two modeling approaches such as local and global modeling. It was first introduced by Gershenfeld et al. (1999). The central idea is to approximate the joint density $p(Y, \mathbf{X})$ of input vectors $\mathbf{X}$ and output variable Y by means of a sum of simple models, each of which contributes to the true density in the vicinity of the cluster center. Once the joint density is successfully built, one is able to make prediction of the new input points. A further research was performed by Ingrassia et al. (2012) in the statistical stand point that presents CWM as a general family of mixture models such as Finite Mixture of Regression Models (FMR)[DeSarbo & Cron (1988); McLachlan & Peel (2000), Fruhwirth-Schnatter (2006)], and Finite Mixture of Regression with concomitant variables (FMRC)[Dayton & Macready (1988); Wedel (2002)]. These *special case* models are unsupervised classification models, since there is no guide from the output vari-

able for correction or calculation of the model error. CWMs have much in common with these models, however, they offer one crucial extension: in each cluster, they introduce a model for the functional dependence called *local model*. This local model creates an interaction between the inputs vector $\mathbf{X}$ and the output variable Y. This interaction implies that CWMs are supervised learners, while Finite Mixture Models are usually unsupervised learners. Finite mixture models are commonly employed in statistical modeling with two different purposes [Titterington et al. (1985)]. In indirect applications, they are used as semiparametric competitors of nonparametric density estimation techniques [McLachlan & Peel (2000); Escobar & West (1995)]. On the other hand, in direct applications finite mixture models are considered as a powerful device for clustering and classification by assuming that each mixture-component represents a group (or cluster) in the original data [Fraley & Raftery (1998); McLachlan & Basford (1988)]. The areas of application of mixture models range from biology and medicine to economics and marketing, [Schlattmann (2009); Wedel & Kamakura (2001)].

A Cluster Weighted Models is a robust model that originated from the finite mixture models. It has been well studied by many researchers. However, there are still areas for improvement. Cluster Weighted Models (CWMs) is an input-output inference framework based on probability density estimation of a joint set of input feature and output target data. It is a flexible technique for approximating an arbitrary function which requires only one hyper-parameter to be fixed beforehand, and provides data parameters such as the length scale (bandwidth) of the local approximation as an output rather than an input of the algorithm.

A broad family of cluster weighted models was introduced in Ingrassia et al. (2015) by assuming that the components conditional distributions belong to the exponential family and the set of covariates were divided into two categories viz; continuous and finite discrete called a mixed-type covariates. Two types of exponential families were considered such as binomial and Poisson cluster weighted models, [Ingrassia et al. (2015)]. An application to real data and a simulation study were carried out. Also, a novel family of twelve mixture models with random covariates was proposed and nested in the linear t CWM, [Ingrassia et al. (2014)], which is more robust than linear Gaussian CWM for heavy long tail data. Solving the inadequacy found with the finite

mixture of linear regressions was the main motivation. The major problem as stated out by Hennig (2000) was the assumption of assignment independence. t CWM has been proposed in the literature [Ingrassia et al. (2012)], where the conditional components were assumed to be linear t regression. The twelve family of models were composed by assuming two distributions; Normal and t distributions with variables and equal constraints imposed on them.

Another innovative generalization of cluster weighted model was proposed to solve the drawbacks of health care effectiveness, [Berta et al. (2016)]. A major drawback addressed was the failure of multilevel model and risk adjustment model (linear and logistic regression) in the presence of data with large unobserved heterogeneity. To overcome this drawback, multilevel cluster weighted model was developed. The main idea of the work was related to multilevel regression mixture model which placed multilevel model in the framework of mixture of regression with fixed covariates, [Muthen & Asparouhov (2009)]. In the same spirit the multilevel model was merged into the mixture of regression with random covariates. The resulting equation below is called Multilevel CWM. In model-based clustering and classification, the cluster-weighted model constitutes a convenient approach when the random vector of interest constitutes a response variable Y and a set p of explanatory variables X, [Subedi et al. (2012)]. The applicability of linear Gaussian CWMs in high dimensional X spaces still remains a challenge, i.e linear Gaussian fails in the presence of high dimensional data. A latent factor structure for X in each mixture component which leads to *cluster-weighted factor analyzers* (CWFA) model was developed to fit high-dimensional data. The extension of linear Gaussian framework to the nonlinear regression in the mixture components has been achieved [Punzo (2014)]. This is useful in modeling data that cannot be captured by linear models or regression. It provided the use which is not restricted to the model-based clustering but also extended to model-based classification. Most of the data in nature are highly skewed which Gaussian distribution is not fit to model. Modeling such data with a Gaussian CWMs can lead to a false grouping. Gutierrez et al. (1995) and Lo et al. (2008) suggested a transformation of skewed data to make it fit for Gaussian components CWMs which in the other hand gives a misleading interpretation. On this note, a model called polynomial Gaussian CWMs to capture the non-elliptical data was proposed. Linear Gaussian CWMs is

the special case of the proposed model. However, the nonlinear extension of linear Gaussian CWM only considered one-dimensional response variables.

Maximum likelihood Estimation (MLE) for CWM is based on Expectation Maximization (EM) algorithm. EM is the standard approach for estimating the parameters of the mixture model, [Dempster et al. (1977)]. EM requires a priori selection of model order such as number of components to be incorporated into the model, and the initialization value. However, EM results depend strongly on the starting values of the parameters. The higher the number of components within the mixture, the higher will be the total log-likelihood. Unfortunately, increasing the number of Gaussians will lead to over-fitting and an increase in the computational burden. Also, EM can be trapped at local maxima and consequently fail to reach global maximal [Wu (1983)]. In a broader view, like many algorithms, EM is a search algorithm that hunts for the best values of the parameters maximizing log-likelihood in a compact space. Additionally, EM does not usually find the best values but seemingly appropriate value in the neighborhood of the parameter space. One simple way that has been explored in the literature is to run the algorithm with different randomly selected starting points and select the starting point that gives the highest likelihood as the global value. This type of solution is computationally expensive because EM algorithm converges slowly. This alone has been a core goal in research, there are many variations of EM to alleviate the problem of local maxima. Multiple restart strategy has been proposed to run EM with multiple random starting points for a specified number of iterations then select the one with highest likelihood and continue from there with the algorithm until convergence, [Hastie & Tibshirani (2004); McLachlan & Peel (2000); Ueda et al. (2000) and Roberts et al. (1998)].

3.3.2 General Formulation

Let $(\mathbf{X}, Y)$ be a pair of random vector $\mathbf{X}$ and random variable Y defined on $\mathcal{D}$ with joint probability $p(\mathbf{x}, y)$, where $\mathbf{X}$ is a d-dimensional input vector with values in some space $\mathcal{X} \subseteq \mathrm{R}^d$ and Y is a response variable having values in $\mathcal{Y} \subseteq \mathrm{R}$. The set of all model parameters is denoted $\boldsymbol{\Theta} = (\omega, \mu, \boldsymbol{\Sigma}, \pi)$. To begin with, we state that $\omega \in \mathrm{R}^{d \times G}$ denotes the weight of the local model to be tuned by stochastic Gradient Descent, location parameter $\boldsymbol{\mu} \in \mathrm{R}^{d \times G}$ where G is the number of groups, $\boldsymbol{\Sigma}$ is the positive

definite covariance matrix, and the π is the mixing distribution with some constraints such as $\sum_g \pi_g = 1$ and $\pi_g > 0$. Since we are dealing with supervised learning, we suppose that for each observation in $(\mathbf{y}, \mathbf{X})$, we have access to the values of the latent variable $\mathbf{Z}$. Now, we have $\{\mathbf{y}, \mathbf{X}, \mathbf{Z}\}$ as our *complete* data set, the set of $\{\mathbf{y}, \mathbf{X}\}$ shall therefore be referred to as *incomplete* data. We stress that in supervised learning, $\mathbf{Z}$ is gotten from $\mathbf{y}$ which contains the position of each observation, and it shall be referred to as *one-hot* encoding in the next chapter.

Generally, CWMs are written as a sum

$$p(\mathbf{x}, y) = \sum_{g=1}^{G} p_g(\mathbf{x}, y), \tag{3.9}$$

where g enumerates the clusters, and $p_g(y, \mathbf{x})$ is a density of a specific form discussed below. The total number of clusters G must be chosen beforehand and can be selected using cross-validation. The density $p_g(\mathbf{x}, y)$ is written as

$$p(\mathbf{y}, \mathbf{x}) \ = \sum_{g=1}^{G} p(\mathbf{y}, \mathbf{x}, z_g) \tag{3.10}$$

The density $p_g(\mathbf{x}, y)$ is written as

$$p_g(\mathbf{x}, y) = \ p(\mathbf{y}|\mathbf{x}, z_g)\ p(\mathbf{x}|z_g)\ \pi_g \tag{3.11}$$

Where $p(z_g) = \pi_g$. The terms in equation (3.11) have the following interpretation:

Cluster Weights: The cluster weight $\pi_g \in [0, 1]$ denotes the amount of data described by the cluster g. The π_g are chosen subject to the constraint

$$\sum_{g=1}^{G} \pi_g = 1 \tag{3.12}$$

Probability of inputs: The density $p_m(\mathbf{x}) = p(\mathbf{x}|z_g)$ describes the domain of influence of cluster g, that is, the distribution of inputs $\mathbf{x}$ around the cluster. They are

chosen with assumption as Gaussian densities, i.e.

$$p(\mathbf{x}|z_g) \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{3.13}$$

$$p_g(\mathbf{x}) = \frac{|\boldsymbol{\Sigma}_g^{-1}|^{1/2}}{(2\pi)^{d/2}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_g) \tag{3.14}$$

with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, effectively describing the location and the range of cluster influence. When working in the high dimensional spaces, it suits well to reduced these input by separable Gaussian, with diagonal matrix of single variances in each dimension, i.e. $\boldsymbol{\Sigma}_g = \text{diag}\ (\sigma_{g,1}, ..., \sigma_{g,d})$.

Output terms: The density $p(y|\boldsymbol{x})$ is the conditional density of the outputs y given the inputs $\boldsymbol{x}$ around the cluster g. The presence of the conditional distribution allows the input vector $\boldsymbol{x}$ to relate with target variable y. In general, they are chosen to as Gaussian densities

$$p(\mathbf{y}|\mathbf{x}, z_g) \sim \mathcal{N}(f(\mathbf{x}, \beta_g), \sigma_g) \tag{3.15}$$

$$p_g(y|\boldsymbol{x}) = (2\pi\sigma_g^2)^{-1/2} \exp(-\frac{1}{2}[y - f(\boldsymbol{x}, \boldsymbol{\beta}_g)]^2/\sigma_g^2) \tag{3.16}$$

with mean $f(\mathbf{x}, \beta_g)$ and variances σ_g^2 describe the local models and the error around the cluster g. The vector $\boldsymbol{\beta}_g$ denote the coefficient of the local model or the weight of contribution associated with the input vector $\boldsymbol{x}$. The $p_g(y|\boldsymbol{x})$ are normalized thus

$$\int p_g(y|\boldsymbol{x})dy = 1 \qquad \forall g, \boldsymbol{x}. \tag{3.17}$$

The cluster functions are chosen based on the type of supervised learning (Regression or classification) we wish to do. It is mostly chosen as linear combination of basis functions $f_i(\mathbf{x})$.

The model output of the CWM is therefore weighted averagely by the local functions $f(\mathbf{x}, \boldsymbol{\beta}_g)$. The Gaussian, which are the input densities $p_g(\mathbf{x})$, controls the behavior of the local functions. The real problem is to find a good parameter values for

- the weights π_g,
- the means $\boldsymbol{\mu}_g$ and the variances $\boldsymbol{\Sigma}_g^2$ of the input density,

- the variances of the output terms σ_g^2 and
- the parameters of the local functions $\boldsymbol{\beta}_g$.

3.3.3 EM algorithm applied to CWMs

In the case of CWMs, as described above in section (3.3) the likelihood becomes easier by introducing a pseudo variable called a latent variable which we can interpret as unobserved data. This unobserved random variable can be imagined as sampling each pair (x_i, y_i) from a single cluster with some probability.

Let $Z_i \in \{1, ..., G\}$ be the label of the cluster that gave rise to (x_i, y_i). This random variable is unobserved. The cluster weights π_g equation (3.11) are interpreted as the probability that $Z_i = g$ for all $g = 1, ..., G$, implying that Z_i are distributed as multinomial distribution parameterized by the cluster weights π_g. Handling cluster model parameters through maximum likelihood would be straight forward if the cluster which generates each sample was known a priori. For example, each cluster center would be the cluster mean of all points from each label. However, since this information is hidden, estimating through maximum likelihood become a nonlinear optimization problem which comes with difficulty. For this problem, EM algorithm is elegant and efficient algorithm when involving latent variables.
The realization of the Z_i are written as an indicator vectors $\boldsymbol{z}_i = (z_{i1}, ..., z_{iG})^T$, where

$$z_{ik} = \begin{cases} 1, & if \quad (\mathbf{x}_i, y_i) \in \boldsymbol{z}_{ik}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

The training set is written as $\Omega_C = \{(\boldsymbol{x}_1, y_1, z_1), ..., (\boldsymbol{x}_N, y_N, z_N)\}$ and the complete log-likelihood $\mathcal{L}_c$ as

$$\mathcal{L}_c(\Omega_C|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \sum_{g=1}^{G} z_{ig} \log p_g(y_i|\mathbf{x}_i) p_g(\mathbf{x}_i) \pi_g, \tag{3.19}$$

where $\boldsymbol{\Theta}$ denotes the entire parameter space of the cluster weighted model, namely the weights, the means $\boldsymbol{\mu}_g$ and the variances $\boldsymbol{\sigma_g}$ of the cluster centers as well as the parameters $\boldsymbol{\beta}_g, \boldsymbol{\Sigma}_g$ of the local models.

The EM algorithm optimization is initialized by the estimates $\boldsymbol{\Theta}^{(0)}$ of these parameters. One possibility, which was also used in the following, is to initialize the cluster weights uniformly i.e. $\pi_g = 1/G$, random cluster mean $\boldsymbol{\mu}_g$ by random numbers or simply picking randomly from the training data and all variances $\boldsymbol{\sigma}_g$ start with identity matrix.
In the expectation step (E-step) of the algorithm as described in chapter (2), the conditional expectation of $\mathcal{L}_c$ is computed with respect to the current parameter estimate, given rise to the following $Q-$ function:

$$Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) = E_{\Theta}\{\mathcal{L}_c(\Omega_c|\boldsymbol{\Theta})|\mathbf{x}, y\} \tag{3.20}$$

The conditional expectation affects only z_{ig} since the terms in the logarithm depend on $\mathbf{x}_i$ and y_i.
E-step is effectively reduced to a calculation of the expectation of z_{ig}, given the observed training data. We introduce

$$q(\mathbf{x}, y; \boldsymbol{\Theta}) = E_{\boldsymbol{\Theta}}(z_{ig}|\mathbf{x}, y) = p(Z_i = g|\mathbf{x}_i, y_i) \tag{3.21}$$

According to the definitions from section (3.3), the posterior probability is in general given by

$$q(\mathbf{x}, y; \hat{\boldsymbol{\Theta}}) = \frac{p_g(y|\mathbf{x})p_g(\mathbf{x})\Theta_g}{\sum_{j=1}^{G} p_j(y, \mathbf{x})} = \frac{p_g(y|\mathbf{x})p_g(\mathbf{x})\Theta_g}{\sum_{j=1}^{G} p_j(y|\mathbf{x})p_j(\mathbf{x})\Theta_g}, \tag{3.22}$$

Each cluster is able to relate with each data point through this distribution. Looking at Equation (3.22), one can see that posterior is the ratio of one cluster to all the the cluster. Given the expectation value, the $Q-$function is given by

$$Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) = \sum_{i=1}^{N}\sum_{g=1}^{G} p_g(\mathbf{x}, y; \hat{\boldsymbol{\Theta}}) \log p_g(y_i|\mathbf{x}_i)p_g(\mathbf{x}_i)\pi_g \tag{3.23}$$

In the maximization step (M-step), the next parameter estimate $\hat{\boldsymbol{\Theta}}$ is obtained by the global maximization of the $Q-$function with respect to $\boldsymbol{\Theta}$ over the parameter space. The derivatives with respect to the desired parameter is calculated by taking the gradient with respect to the parameter of interest and setting to zero, thus obtain a new set of parameters $\boldsymbol{\Theta}$ as a function of the old parameters $\hat{\boldsymbol{\Theta}}$. This procedure is repeated until convergence.
Applying the logarithmic law, $Q-$function can be decomposed as follows:

$$\begin{aligned} Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) = & \sum_{i=1}^{N} \sum_{g=1}^{G} p_g(\mathbf{x}, y; \hat{\boldsymbol{\Theta}}) \log p_g(y_i|\mathbf{x}_i) \\ & + \sum_{i=1}^{N} \sum_{g=1}^{G} p_g(\mathbf{x}, y; \hat{\boldsymbol{\Theta}}) \log p_g(\mathbf{x}_i) \\ & + \sum_{i=1}^{N} \sum_{g=1}^{G} p_g(\mathbf{x}, y; \hat{\boldsymbol{\Theta}}) \log \pi_g \end{aligned} \tag{3.24}$$

This decomposition is useful as taking the gradient with respect to the parameter of interest becomes convenient. For example, the cluster weights π_g, can be computed independently of the other while others summands without the parameter of interest becomes zero automatically. Since the weights are with constraints $\sum \pi_g = 1$ and $0 \leq \pi_g \leq 1$, Lagrange multiplier is introduced as follows:

$$\frac{\partial}{\partial \pi_g} \left[Q(\Theta; \hat{\boldsymbol{\Theta}}) + \lambda (1 - \sum_{g=1}^{G} \pi_g) \right] = 0, \tag{3.25}$$

which, now leads to

$$\pi_g = \frac{1}{N} \sum_{i=1}^{N} p_g(y_i, \mathbf{x}_i; \hat{\boldsymbol{\Theta}})$$

which can equally be interpreted as $\sum_i z_{ig}/N$, where the unknown labels are substituted by their expectation value.

The update estimates for the means and variances of the clusters $(\boldsymbol{\mu}_g, \boldsymbol{\sigma}_g)$ are also derived by maximized $Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}})$. Thus, the updated means are given by

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{N} \mathbf{x}_i p_g(y_i, \mathbf{x}_i; \hat{\boldsymbol{\Theta}})}{\sum_{i=1}^{N} p_g(y_i, \mathbf{x}_i; \hat{\boldsymbol{\Theta}})} \tag{3.26}$$

3.3.4 Geometrically Constrained CWMs

The full multivariate Gaussian for CWMs discussed above has posed a lot of problem in the estimation process. Some of the problems which are due to high-dimensional space or large d can be associated to the problem of matrix inversion caused by singularity, degeneracies of the algorithm. For full covariance matrix the parameters to be estimated are $(G-1) + Gd + G[d(d+1)/2]$. This parameters are quite a large number number. For example in the Epileptic Seizure data, with $d = 178$ and $G = 5$, this is $128,879$ parameters to be estimated, which is too large for any clustering model. Such a large numbers of parameters can lead to difficulties in estimation, including lack of precision or even cause the algorithm to degenerate. They also reduce the computational speed of the algorithms. In order to mitigate this problem, Banfield & Raftery (1993) and Celeux & Govaert (1995) introduced the eigenvalue decomposition of the cluster covariance matrix Σ_g, in the form

$$\Sigma_g = \lambda_g D_g A_g D_g^T. \tag{3.27}$$

In Equation (3.27), D_g is the matrix of the eigenvectors of Σ_g, $A_g = diag\{A_{1,g}, ..., A_{d,g}\}$ is a diagonal matrix whose elements are proportional to the eigenvalues of Σ_g arranged in a descending order, and λ_g is the constant associated with the proportionality.
Each elements in this decomposition corresponds to a particular geometric property of the gth component. The matrix of the eigenvectors D_g determines its orientation in R^d. The diagonal matrix of scaled eigenvalues A_g governs its shape. The region where the gth is densely concentrated can be determined by the maximum number of the shape in the plane. For example, if $A_{1,g} \gg A_{2,g}$, then the gth component is tightly concentrated around a line in R^d. If $A_{1,g} \approx A_{2,g} \gg A_{3,g}$, then the gth component is concentrated in a two-dimensional plane in R_d. If all the values of $A_{j,g}$ are approximately equal, then the gth component is roughly equal. The constant

of proportionality determines the volume. This is proportional to $\lambda_g^d|A_g|$ where $||A_g$ is determinant of A_g preferably constrained to be equal to 1. Parsimony occurs in different ways using the decomposition by either constraining any or all of the volume, shape or orientation to be to be equal or varied across the clusters. Also, the covariance matrix can be forced to be spherical i.e. Identity matrix I. Whenever the covariance matrix is spherical, there are two univariate models, and 14 possible models in multivariate case.

Table 3.1: Parameterizations of the covariance matrix Σ_g through Eigenvalue decomposition. A denotes a diagonal matrix

Identifier	Model	Distribution	Volume	Shape	Orientation
E	–	Univariate	Equal	Not required	Not required
V	–	Univariate	Variable	Not required	Not required
EII	λI	Spherical	Equal	Equal	Not required
VII	$\lambda_g I$	Spherical	Variable	Equal	Not required
EEI	λA	Diagonal	Equal	Equal	Axis-aligned
VEI	$\lambda_g A$	Diagonal	Variable	Equal	Axis-aligned
EVI	λA_g	Diagonal	Equal	Variable	Axis-aligned
VVI	$\lambda_g A_g$	Diagonal	Variable	Variable	Axis-aligned
EEE	Σ	Ellipsoidal	Equal	Equal	Equal
VEE	$\lambda_g DAD^T$	Ellipsoidal	Variable	Equal	Equal
EVE	$\lambda DA_g D^T$	Ellipsoidal	Equal	Variable	Equal
EEV	$\lambda D_g AD_g^T$	Ellipsoidal	Equal	Equal	Variable
VVE	$\lambda_g DA_g D^T$	Ellipsoidal	Variable	Variable	Equal
VEV	$\lambda_g D_g AD_g^T$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda D_g A_g D_g^T$	Ellipsoidal	Equal	Variable	Variable
VVV	Σ_g	Ellipsoidal	Variable	Variable	Variable

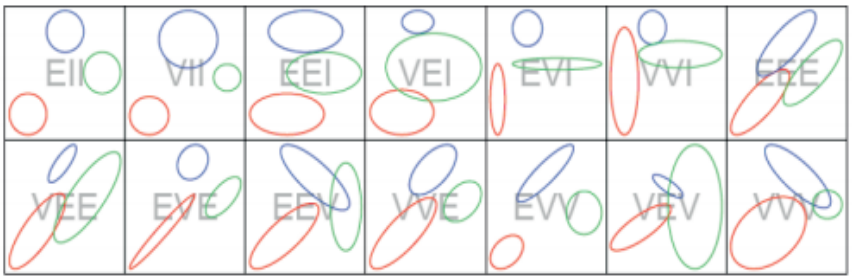


Figure 3.1: Models Used in CWMs clustering: Example of contours of the bivariates normal component densities for the 14 parameterization of the covariance matrix. Source: Bouveyron et al. (2019)

Table (3.1) shows the multivariate models denoted by three-letter identifier where "E" stands for Equal and "V" stands for variable. If the first letter is "E" it means the volume is equal/constant across the clusters, and "V" if varied across. In the same vein, the second letter "E" represents equal shape and "V" if not, so that for all $g = 1, ..., G$, the shape matrices $A_g \equiv A$. "I" stands for spherical when the $A_g = I$ for $g = 1, ..., G$. Finally, if "E" is located at the the third position, then the D_g of eigenvectors specify the cluster orientations are equal $D_g \equiv D$ for $g = 1, ..., G$, "V" if they are not constrained, and "I" if the clusters are spherical such that $D_g = I$ for $g = 1, .., G$.

Figure (3.1) shows the examples of contours of the component densities for the various models in the two-dimensional case with two mixture components. These constrained models can have extremely fewer parameters that need to be estimated independently than the full covariance model, while fitting the sample data almost as well. The constrained models can yield more precise estimates of model parameters, accurate out-of-sample predictions, and easy interpretability of parameter estimates. Moreover, the model have Gd parameters for the component means $\boldsymbol{\mu}_g$, and $(G-1)$ parameters for the mixture proportions π_g.

Table (3.2) shows the numbers of parameters needed to specify the covariance matrix for each model in the 178-dimensional five-component case, $d = 178, G = 5$, three-dimensional five-component case, $d = 3, G = 5$ gotten from the Epileptic seizure recognition data before dimensionality reduction, and after dimensionality reduction, respectively. Before performing the dimensionality reduction, we note that CWM is

impracticable. These results are obtained by noting that for one mixture component, the volume is specified by 1 parameter, the shape by $(d-1)$ parameters, and the orientation by $d(d-1)/2$. The potential gain in the combination of parsimony and dimensional reduction is far higher than the gain achieved from only parsimony compared to the full covariance matrix parameters.

Table 3.2: Numbers of the parameters needed to specify the covariance matrix for models used CWMs and CWMs-tSNE

Model	General	$d = 3, G = 5$	$d = 178, G = 5$
E	–	–	–
V	–	–	–
EII	1	1	1
VII	G	5	5
EEI	d	3	178
VEI	$G + (d-1)$	7	182
EVI	$1 + G(d-1)$	11	886
VVI	Gd	15	890
EEE	$d(d+1)/2$	6	15931
VEE	$G + (d+2)(d-1)/2$	10	15935
EVE	$1 + (d+2G)(d-1)/2$	14	16639
EEV	$1 + (d-1) + G[d(d-1)/2]$	18	78943
VVE	$G + (d+2G)(d-1)/2$	18	16643
VEV	$G + (d-1) + G[d(d-1)/2]$	22	78947
EVV	$1 + G(d+2)(d-1)/2$	26	79651
VVV	$G[d(d+1)/2]$	30	79655

In the most extreme case in Table (3.2), in the 178-dimensional case with 5 mixture components, the VVV model requires $79,655$ parameters to represent the covariance matrices, whereas the same VVV requires 30 parameters with the combination of dimensionality reduction and eigenvalue decomposition. Although, there are some

gains in parsimony, however it has been observed that the most parsimonious models do not always fit the data adequately. Moreover, the number of parameters to be estimated in parsimonious model is still outrageously high, and the preferable solution would be to apply some steps further parsimonious method to the results of the parsimonious model. However, this might not achievable if the computational time is a priority. Alternatively, the best solution would be to perform dimensionality reduction before using parsimony.

Unfortunately, Eigenvalue decomposition method does what we can call a "local parameter reduction" when the "global feature" remains huge. Fitting the huge original high-dimensional data irrespective of the parsimony encumbers CWMs model. Consequentially, reducing the classification power, slows the computation speed, and lead to misinterpretation of the result. This becomes a challenge in CWMs models.

3.3.5 The theory of tSNE

The Stochastic Neighboring Embedding (SNE) was first introduced by Hinton & Roweis (2002). SNE aims to place the objects in a low-dimensional space in order to retain neighboring identity, and can be naturally extend to allow multiple different low-dimensional images of each object, [Hinton & Roweis (2002)]. As a dimensional reduction technique, SNE can construct a reasonably good performance of visualizations, however, it is hindered by a complex cost function that is difficult to optimize. Maaten & Hinton (2008) introduced a variation of SNE called t Distributed Stochastic Neighbor Embedding (tSNE). The aim of tSNE is to transform the high-dimensional data set X $= (x_1, ..., x_n)$ into low-dimensional data set Y $= (y_1, ..., y_n)$. tSNE is much easier to easier to optimize, and provides significantly better visualization by reducing the tendency to crowd points together in the center of the map, [Maaten & Hinton (2008)]. The cost function employed in tSNE is difference from that of SNE. tSNE employed a symmetric version of SNE as an alternative to mitigate the problem of the presence of outliers. The asymmetric SNE used in SNE is given as follows;

$$q_{ij} = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq i} exp(-||y_k - y_j||^2)} \tag{3.28}$$

where q_{ij} is the pairwise similarities in the in the low dimensional map and the way to define the pairwise similarities in the high-dimensional space p_{ij} is given by

$$p_{ij} = \frac{exp(-||x_i - x_j||^2)/2\sigma^2}{\sum_{k \neq i} exp(-||x_k - x_j||^2/2\sigma^2)} \tag{3.29}$$

These equations are referred to as symmetric because it has the that $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$. Another uniqueness with tSNE is that tSNE applies a Student t-distribution with one degree of freedom similar to Cauchy distribution as the heavy-tailed distribution in the low-dimensional space. The joint probabilities for the low-dimensional map q_{ij} instead becomes

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k \neq i} \left(1 + ||y_i - y_j||^2\right)^{-1}} \tag{3.30}$$

The advantages of employing a Student t-distribution can be found in Maaten & Hinton (2008). The ultimate goal of t-SNE is to represent p_{ij} by $q_i j$ as accurate as possible, so the cost function is given by

$$C = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{3.31}$$

Gradient method is introduced for minimizing the cost function and the gradient has the form given by

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + ||y_i - y_j||^2\right)^{-1} \tag{3.32}$$

Equation (3.32) can be interpreted as the summation of a resultant force pulling y_i in the direction of y_j or pushing it away depending on whether j is observed as a neighbor of i. The gradient descent is initialized by sampling the map point $\mathrm{Y}^{(0)} = (y_1, ...y_n)$ randomly from $\mathcal{N}(0, 10^{-4}I)$. A momentum is added to the gradient descent to speed up the optimization and avoid being stuck in local optimal. Finally, the gradient update is given by

$$\mathrm{Y}^{(t)} = \mathrm{Y}^{(t-1)} + \zeta \frac{\partial C}{\partial \mathrm{Y}} + \alpha(t)(\mathrm{Y}^{(t-1)} - \mathrm{Y}^{(t-2)}) \tag{3.33}$$

where $Y^{(t)}$ is the solution at the iteration t, ζ is the learning rate, and the $\alpha(t)$ is the momentum at iteration t.

3.3.6 Dimensionality reduction

Given the high-dimensional nature of the dataset considered here, a preprocessing step of feature extraction is of great importance to reduce the computational burden and time complexity before fitting the CWMs model. The considered preprocessing step proceeds as follows; first of, we fit the feature set to the tSNE, and afterwards we project the test unit nonlinearly to obtain the low-dimensional subspace. Without dimensionality reduction process, CWMs can be so limited by the high-dimensional data which slows down the computational speed and hamper the clustering performance of the model. The subspace of the original features are then filtered into the CWMs for clustering analysis. Moreover, the visualization of the high-dimensional data is made possible by tSNE technique. Here, we present both low-dimensional data and high-dimensional data with features running to the order of hundred.

3.4 Application to real data

This section illustrates some real data applications of the linear CWMs defined above with a substantive high dimensionality. The analysis is performed using the **R** package for CWMs called **FlexCWM**, [Mazza et al. (2018)].

3.4.1 Abalone data

The first application concerns the prediction of age of abalone from physical measurements. The data was taken from UCI Repository (UCI) database with the original sources of Marine Resources Division and Sam (1995); Warwick et al. (1994). The age of abalone was determined by counting the number of rings through a microscope after cutting the shell through the cone, and staining it. The analysis presented below uses all the variables in the dataset. The following are the attributes of the data;
Sex: Male (M), Female (F), and Infant (I), Length: Longest shell measurement, Diameter: Perpendicular to length, Height: With meat in shell, Whole.Weight: Whole

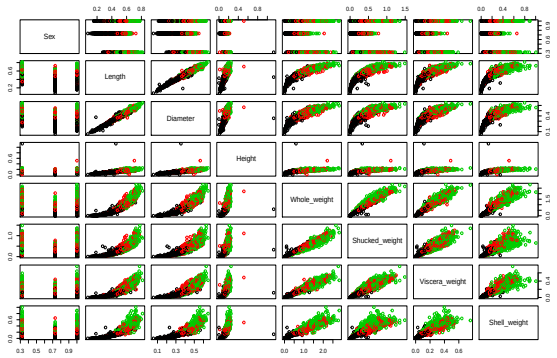


Figure 3.2: The visualization of the descriptive summary of the original Abalone data colored according to the grouping of the Rings: Black $(1-8)$, Red $(9-10)$, and Green (≥ 11)

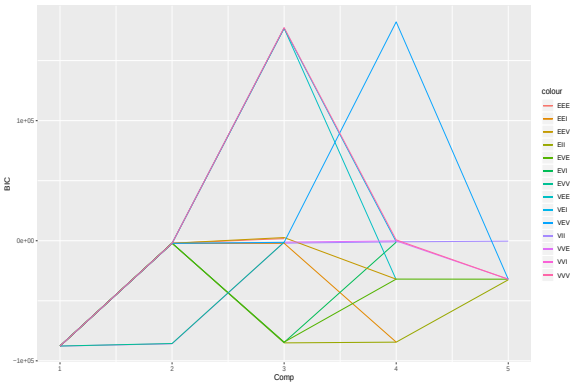


Figure 3.3: Model selection for the Abalone data using BIC values of the fourteen models. The BIC produced by three models select the correct number of components.

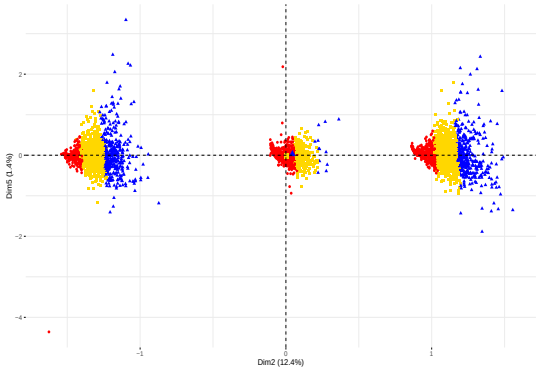


Figure 3.4: The classification plot of CWM-tSNE for $G = 3$ with model VVV as selected by BIC.

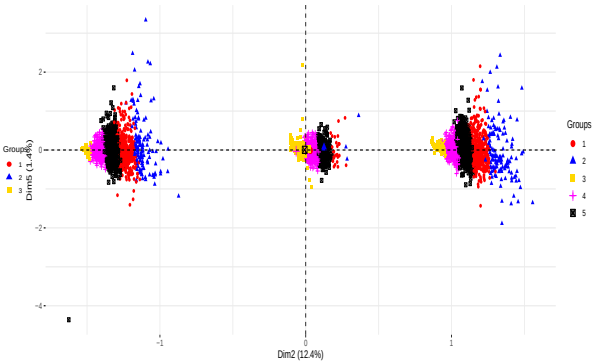


Figure 3.5: The classification plot of CWM-tSNE for $G = 4$ as suggested by the BIC

abalone, Shucked.Weight: Weight of meat, Viscera.Weight: Gut weight after bleeding, Shell.Weight: Gut weight after being dried, Rings: Age in years of the Abalone. There are $G = 3$ groups of abalone with respect to Sex variables: $M = 1528$, $F = 1307$, and $I = 1342$. First off, we use the whole variables and check the effect on the

clustering power of the linear CWM. We compare the Bayesian Information criteria (BIC) produced by fourteen different parsimonious models.

Figure (3.2) concerns the observed labeled data. This graphical representation is the visualization of the descriptive summary of the abalone data. The observations are color-coded according to the group of the Rings variable grouped into 3- class category; $1-8$, $9-10$, and ≥ 11. The goal is to classify the abalone according to their age group. The nonlinear projection from the original feature space to low dimensional feature space is performed. However, the goal is not to separate the observation to their respective classes but to reduce the dimension of the data which leads to the removal of any multi-collineariy among the features. Afterwards, we filtered the projected feature into CWMs. This is always better in terms of speed and accuracy. We perform the analysis on the original data and the parsimonious models selected the same number of component as the projected data. This assures us that the low-dimensional data is a good representation of the original data. However, all the eight information criteria have an extremely high number produced by the original data. This might be due to redundancy in the feature of the original data.

The data can be seen as a nested cluster or as having both global and local components, i.e. cluster through the sex variable of the Abalone which are male (M), Infant (I), and Female (F), and the grouping through the age of the Abalone. This makes the data extreme difficult to separate. The previous work by Sam (1995) also confirmed the presence of overlap in the data while he suggested additional information to separate the class completely using the affine combinations. We note that it is easier to separate the data with respect to the sex variable while the age group remains cluttered together. This can hinder the performance of the clustering algorithm.

Table 3.3: The selection of the best model among 14 models according to the BIC is VVV

Model	comp1	comp2	comp3	comp4	comp5
EII	-87696.1	**-2109.2**	-85062.3	-84460.1	-32541.1
VII	-87696.1	-2230.8	-2024.8	-917.4	**-341.9**
EEI	-87688.0	-2115.9	**-2101.1**	-84412.6	Not Estimated
VEI	-87688.0	-2184.1	**176940.8**	-838.7	Not Estimated
EVI	-87688.0	-2006.1	-84632.3	**-1225.5**	Not Estimated
VVI	-87688.0	-2085.3	**177254.5**	Not Estimated	-32220.5
EEE	-87687.8	-2122.8	**1793.2**	Not Estimated	-32554.9
VEE	-87687.8	-2146.5	**176932.7**	-32459.0	Not Estimated
EVE	-87712.7	**-1999.1**	-84415.3	-31949.4	-32016.0
EEV	-87687.8	-2086.2	**2651.1**	-32021.3	Not Estimated
VVE	-87734.3	-85777.6	-1219.1	**-118.8**	-32002.6
VEV	-87687.8	-2186.5	-1337.2	**182260.9**	-32072.1
EVV	-87687.8	-85606.0	-1196.0	Not Estimated	**-1063.1**
VVV	-87687.8	-2108.3	**177462.9**	715	-32028.74

Figure (3.3) shows the values of BIC for the models in the CWMs-tSNE with G ranging from $1, ..., 5$. We show the plot resulting from BIC. In CWMs-tSNE model, the four models that provide the largest values for the BIC were VEI, VVI, VEE, VVV with values: $176940.8, 177254.5, 176932.7$, and 177462.9. In Table (3.3), we presented only the BIC values for the 14 models considered because the eight information criteria agreed in selecting the same number of components. The best models are distinguished with boldface.

Also, the ARI and its variants are presented in Table (3.4). The ARI for the models selected by the BIC as shown in Table (3.3) is 1. In contrast, according to Sam (1995) the Cascading-Correlation with no hidden nodes and with 5 hidden nodes had 24.8% and 26.2% , while C4.5 achieved 21.5%, Linear Discriminant Analysis (LDA) achieved 0.0%, and the $k = 5$ Nearest Neighbor got 3.57% accuracy.

Table 3.4: Adjustment Rand Index and its variants of the three-component Model for Abalone data. According to the BIC, the models VEI, VVI, VEE, and VVV give ARI = 1

Model	Rand	HA	MA	FM	Jaccard
EII	0.780	0.576	0.576	0.777	0.603
VII	0.782	0.471	0.471	0.627	0.447
EEI	0.822	0.599	0.599	0.733	0.577
VEI	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
EVI	0.794	0.513	0.513	0.662	0.490
VVI	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
EEE	0.823	0.603	0.603	0.735	0.582
VEE	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
EVE	0.781	0.576	0.576	0.777	0.603
EEV	0.809	0.571	0.571	0.715	0.556
VVE	0.782	0.488	0.488	0.647	0.475
VEV	0.957	0.901	0.901	0.934	0.872
EVV	0.798	0.519	0.520	0.666	0.493
VVV	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

3.4.2 Protein data

The goal of the second application is to cluster the localization site of proteins. The protein data created by Horton & Nakai (1996) and is available in the UCI database. The data consist of seven input variables and class variable. There are $N = 336$ observations and attributes information is as follows;

Sequence Name: Accession number for the SWISS-PORT database, mcg: McGeoh's method for signal sequence recognition, gvh: Von Heijne's method for signal sequence recognition, lip: von Heijne's signal Peptidase II consensus sequence score, chg: Presence of charge on N-terminus of predicted lipoproteins, aac: Score of discriminant analysis of the amino acid content of outer membrane, alm1: Score of the ALOM membrane spanning region prediction program, alm2: Score of ALOM program after excluding putative cleavable signal regions from the sequence. According to the framework of CWMs, we transformed the multiclass response called the localized site by adding the 0.5 and taking the logarithm of the result. This is done to transform from a categorical variable to continuous. We pretended as if the true clustering is not known apriori and check which model would perform the best among the fourteen

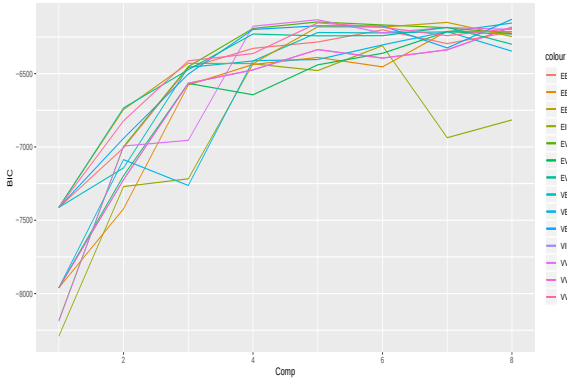


Figure 3.6: Model selection for the protein data using BIC values of the fourteen models. The BIC produced by five models select the correct number of components.

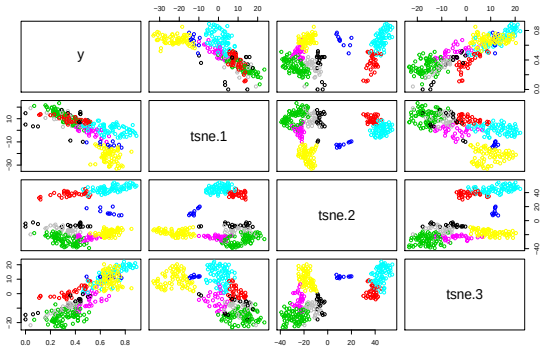


Figure 3.7: The plot produced by CWMs after dimension reduction via tSNE. CWMs selected eight components which aligns to the true class of the localization site of protein.

parsimonious models. In order to visualize the BIC values, Figure (3.6) shows the BIC plot for the protein data, using the **R** commands provided by the **FlexCWM** package. Values are shown for up to $G_{max} = 8$ components and for the 14 covariance models estimated in the same package, i.e. for 8×14 different competing models in all. BIC selects the model with six mixture components and the EEE with 10 other covariance specifications, in which all the covariance matrices are either equal or varied. However, BIC selects the five models such as VII, VVI, VEE, and VEV with eight mixture components. Table (3.5) lists the values of the BIC for the fourteen models. The values of the BIC according to the Table (3.5) are -6183.0 (VII), -6183.0 (VVI), -6155.1 (VEE), and -6129.2 (VEV). Among all the models considered, the value of the BIC -6309.6 produced by EII is the worst model. Table (3.6) is generated by comparing the clusters produced by the BIC values with the true class of the localized site using the varieties of ARI. VVE model shows higher values of the ARI among all the models. According to the selection of the component produced by the CWM-tSNE model, Figure (3.6) shows the classification for the VVI selected model with respect to the number of cluster produced by the CWMs-tSNE. We note that AWE gives wrong number of clusters throughout the analysis. The protein data has been analyzed by Horton & Nakai (1996). In their work of "A Probabilistic Classification System for predicting the Cellular Localization Sites of Proteins", their model

Table 3.5: The comparison of the BIC produced by the fourteen parsimonious models after performing the dimensionality reduction

Model	comp1	comp2	comp3	comp4	comp5	comp6	comp7	comp8
EII	-8291.1	-7270.3	-7217.3	-6433.0	-6479.2	**-6309.6**	-6937.0	-6815.8
VII	-7960.9	-7223.6	-6565.9	-6474.5	-6336.2	-6394.1	-6337.5	**-6183.0**
EEI	-7960.9	-7422.0	-6577.6	-6437.5	-6389.8	-6453.8	**-6210.9**	-6213.5
VEI	-7960.9	-7086.6	-7262.7	-6411.8	-6404.0	-6304.2	**-6213.3**	-6346.9
EVI	-7960.9	-7196.4	-6566.5	-6644.1	-6440.1	-6361.3	**-6214.6**	-6230.0
VVI	-7960.9	-7223.6	-6565.9	-6474.5	-6336.2	-6394.1	-6337.5	**-6183.0**
EEE	-7412.5	-7005.6	-6460.8	-6327.7	-6284.3	**-6201.5**	-6295.4	-6214.2
VEE	-7412.5	-7143.0	-6457.2	-6413.9	-6219.8	-6222.3	-6215.4	**-6155.1**
EVE	-8182.5	-6997.2	-6452.2	-6192.0	**-6147.5**	-6168.1	-6186.7	-6227.5
EEV	-7412.5	-6745.1	-6429.6	-6431.5	-6180.8	-6182.5	**-6150.6**	-6247.5
VVE	-8189.6	-6994.1	-6954.8	-6176.8	**-6132.6**	-6222.1	-6185.7	-6198.5
VEV	-7412.5	-6941.4	-6503.3	-6199.0	-6174.0	-6174.7	-6324.7	**-6129.2**
EVV	-7412.5	-6733.9	-6473.3	-6229.9	-6242.5	-6241.3	**-6184.3**	-6299.9
VVV	-7412.5	-6822.0	-6412.8	-6363.1	**-6154.2**	-6183.6	-6241.4	-6192.4

Table 3.6: Adjustment Rand Index and its variants of the fourteen parsimonious models to select the hidden structure or cluster in the protein data

Model	Rand	HA	MA	FM	Jaccard
EII	0.821	0.478	0.483	0.603	0.411
VII	0.794	0.429	0.435	0.567	0.389
EEI	0.799	0.413	0.419	0.549	0.360
VEI	0.800	0.428	0.434	0.562	0.378
EVI	0.777	0.336	0.343	0.484	0.299
VVI	0.798	0.414	0.420	0.550	0.364
EEE	0.799	0.412	0.419	0.549	0.359
VEE	0.848	0.587	0.591	0.690	0.522
EVE	0.816	0.468	0.474	0.594	0.405
EEV	0.788	0.409	0.415	0.551	0.373
VVE	**0.866**	**0.646**	**0.649**	**0.737**	**0.581**
VEV	0.803	0.446	0.452	0.578	0.396
EVV	0.788	0.406	0.411	0.547	0.368
VVV	0.783	0.372	0.379	0.516	0.334

achieved 81% classification accuracy. Also similar accuracy has been achieved for Binary Decision Tree and Bayesian Classification methods.

3.4.3 Epileptic Seizure Recognition

We now analyze the Epileptic Seizure recognition data gotten from UCI. The original dataset consists of 5 different folders, each with 100 files, with each file representing a single subject/person. Each file is a recording of brain activity for 2.36 seconds. The corresponding time-series is sampled into 4097 data points. Each data point is the value of the EEG recording at a different point in time. So there is a total of 500 individuals with each having 4097 data points for 23.5 seconds. Every 4097 data points is divided and shuffled into 23 chunks, and each chunk contains 178 data points for 1 second. Each data point is the value of the EEG recording at a different point in time. So there is a total of 11500 pieces of information (row), each with 178 data points for 1 second (column), then the last column represents the class $y = \{1, 2, 3, 4, 5\}$. The Epileptic data contains $178-$dimensional input vector. The dependent variable y is defined as follows; 5: eyes open when the EEG signal of the brain was recorded. 4: means eyes closed when the EEG signal was recorded, 3: mean they identified where the region of the tumor was in the brain and the recorded the EEG activity from the healthy brain area, 2: means the EEG was recorded from the area where the tumor was located, and 1: means the recording of seizure activities.

The goal is to detect the underlying component of the data. In the previous works, the data has been treated as a binary classification where class 1 represents the presence of seizure in a patient and $2, 3, 4, 5$ represent the absence of seizure. The label class is distributed equally as 2300.

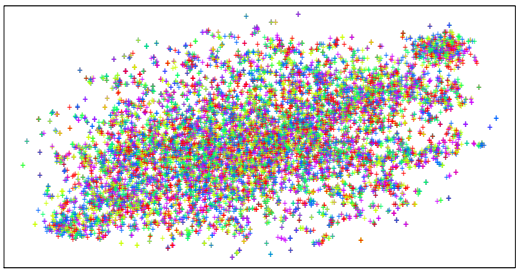


Figure 3.8: The tSNE for dimensionality reduction of the Epileptic Seizure data for 1000 iteration, perplexity = 15 and theta = 0.5.

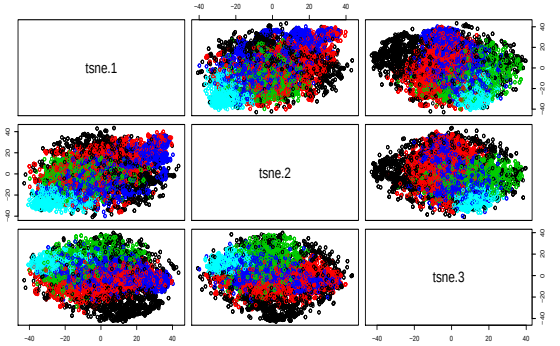


Figure 3.9: The CWM-tSNE plot for clustering the low-dimensional data produced by tSNE for Seizure recognition data with five categories.

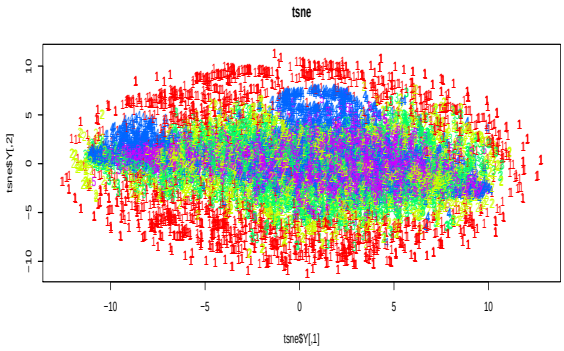


Figure 3.10: The tSNE for dimensionality reduction of the Epileptic Seizure data for $10,000$ iterations, perplexity = 250, and theta = 0.5.

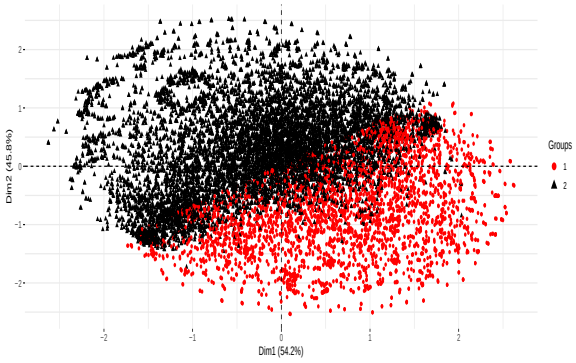


Figure 3.11: The CWM-tSNE plot for clustering the low-dimensional representation of Seizure recognition data produced by tSNE with EEE model.

The Cluster Weighted Models (CWMs) employs the Ordinary Least Squares (OLS) for its maximization step of the EM algorithm, therefore it becomes inappropriate to fit the dependent variable which is a categorical variable. An alternative approach is to take the logarithm of the label class and add some noise to make it a continuous variable. Afterwards, we performed the dimensionality reduction on the independent

variable of order 178. We note here again that the goal of tSNE is not for clustering, however we prioritize dimensionality reduction over clustering with tSNE.

Figure (3.8) visualizes the high-dimensional data on a $2D$ plane with the perplexity = 15, iteration = 1000 and the theta = 0.5. According to the plot shown in Figure (3.8), there is a linear pattern as revealed by the tSNE. We observed that when the perplexity is between 9 and 15, and the theta = 0.5, tSNE gives an unsatisfactory low-dimensional data, this is called a "crowd point". However, due to high volume of the data, tSNE tends to be a bit slower than when performed on a moderately high-dimensional data. According to the setup of tSNE, there is a trade-off between speed and accuracy. The hidden structure in the high-dimensional data is preserved in the low-dimensional space. However, the epileptic seizure data is highly overlapped, this makes clustering extremely difficult to perform. Figure (3.9) shows a five-component structure of the CWMs plot on the low-dimensional data filtered into the CWMs model. Almost all the information criteria selected model with 5 mixture components. Although, we are able to visualize the high-dimensional data but the clusters are not well separated. One limitation associated with the tSNE output in Figure (3.8) is that the information criteria tend to favor the number of the label class. This is however contrary to previous works which have performed binary classification where class 1 represent presence of Epileptic seizure in the patients against the absence of Epileptic seizure. To reduce the crowd points in Figure (3.8), we further performed a thorough dimensionality reduction with different parameters of the tSNE; the Perp = 250, theta = 15, with $10,000$ iterations. The output after $10,000$ iterations is presented in Figure (3.10).

From the plot in Figure (3.10), the underlying structure was revealed after $10,000$ iterations but tSNE alone is not strong enough to cluster the label class into two classes. CWM-tSNE however worked on the output of the tSNE to reveal the hidden 2-categorical structure in the seizure data. The plot of CWM-tSNE is shown in Figure (3.11). This is the plot produced by the model EEE selected by ICL. CWM-tSNE produces a distinct two classes but with some misclassifications. In Figure (3.11), 1 represents the presence of Epileptic seizure and 2 represents the absence of the Epileptic seizure. The number of components selected by BIC does not agree with one selected by ICL when using the model EEE. BIC suggested that the number of

Table 3.7: The comparison of the BIC and ICL produced by the fourteen parsimonious models after performing the dimensionality reduction

Model	comp1	comp2	comp3	comp1	comp2	comp3
EII	-169881	-166546	-163231	-169881	-169186	-164750
VII	-169881	-166556	-162870	-169881	-169185	-164866
EEI	-168667	-164041	-162360	-168667	-165176	-165164
VEI	-168667	-164050	-162591	-168667	-165176	-164680
EVI	-168667	-164050	-160881	-168667	-165185	-162118
VVI	-168667	-164059	-159935	-168667	-165183	-161259
EEE	-168593	-163889	-163367	-168593	**-165081**	-166885
VEE	-168593	-163898	-162569	-168593	-165090	-164475
EVE	-169086	-164816	-161938	-169086	-166081	-163565
EEV	-168593	-163898	-162271	-168593	-165090	-163699
VVE	-169109	-163927	-159226	-169109	-165145	-160531
VEV	-168593	-163908	-162034	-168593	-165099	-163612
EVV	-168593	-163908	-161937	-168593	-165103	-163613
VVV	-168593	-163149	-159204	-168593	-164215	-160499

Table 3.8: Adjustment Rand Index and its variants of the fourteen parsimonious models to select the hidden structure or cluster in the protein data

Model	Rand	HA	MA	FM	Jaccard
EII	0.557	0.153	0.153	0.618	0.432
VII	0.477	0.052	0.053	0.520	0.328
EEI	0.503	0.083	0.083	0.555	0.363
VEI	0.474	0.053	0.053	0.516	0.323
EVI	0.708	0.428	0.428	0.759	0.596
VVI	0.486	0.071	0.071	0.529	0.336
EEE	0.601	0.152	0.152	0.686	0.520
VEE	0.478	0.056	0.056	0.522	0.329
EVE	0.712	0.434	0.434	0.763	0.601
EEV	0.539	0.152	0.153	0.589	0.394
VVE	0.486	0.069	0.069	0.529	0.336
VEV	0.509	0.106	0.107	0.556	0.360
EVV	0.705	0.421	0.421	0.756	0.592
VVV	0.485	0.068	0.068	0.529	0.336

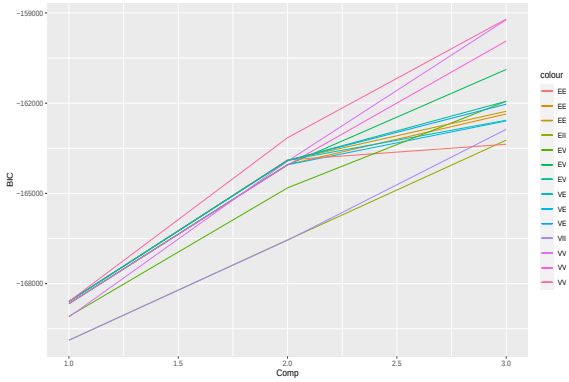


Figure 3.12: The Model selection of BIC for Seizure data among the fourteen parsimonious model; BIC selected wrong number of mixture component when the true component according to the label is two-categorical.

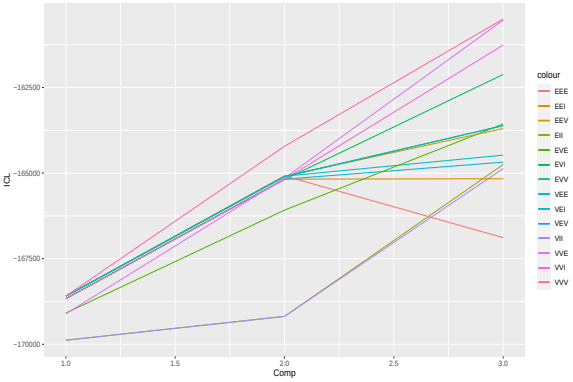


Figure 3.13: The Model selection of ICL for Seizure data among the fourteen parsimonious model; ICL selected EEE with the correct number of mixture component when the true component according to the label is two-categorical.

components is 3, while ICL suggested that the hidden number of components is 2. In the other models, the number of components selected by BIC agreed with ICL as they all selected 3 mixture components. Figure (3.12) and Figure (3.13) show the comparison between BIC and ICL on the number of mixture components. The values are provided in the Table (3.7). The left values are produced by BIC and the right values are the ICL. The ARI and its variants are provided in Table (3.8). The model with the highest values of ARI is EVE model. However, the classification accuracy produced by EEE model is 73%.

3.5 Summary

In this chapter, we investigated the use of CWMs model on moderately high dimensional and extremely high-dimensional data. First, we reviewed the general background study of the CWMs according to Ingrassia et al. (2014) and explained how they metamorphosed from a finite mixture model (FMM). According to Hennig (2000), the

problem associated with FMM is the assumption of assignment independence, i.e. the assignment of the data points to the cluster has to be independent of the covariates.

On the contrary, CWMs assume random covariates with a parametric specification which allows for assignment dependence. We further derived the EM algorithm for the parameter estimations. The limitations of CWMs are the main motivation of the chapter. The limitation of CWMs is the effect of the "curse of dimensionality". The clustering performance of CWMs is hampered by the dimensionality of the data. However, the eigenvalue decomposition only has a little improvement in the face of huge high-dimensional data. For example, the seizure data of 178 dimensions has $128,879$ parameters to estimate. This may be impractically attainable in real time when using CWMs, unlike RandomForest that performs internal feature selection. However, the use of eigenvalue decomposition only solves the problems in-part by a little reduction in the number of parameters to be estimated. In the presence of high-dimensional data with CMWs, denegeracies are inevitable, misinterpretation is bound to occur, the computation time increases proportionally with the dimensionality of the data, and low classification performance. For example, an original CWMs fails to cluster an image data with 784-dimensions.

To alleviate these limitations in CWMs, we introduce a CWMs based on tSNE for high-dimensional data. tSNE is a very powerful dimensionality reduction technique introduced by Maaten & Hinton (2008). We first performed a dimensionality reduction based on different parameters of **Rtsne** package in **R**. The approach called CWMs-tSNE is applied to real high-dimensional Epileptic Seizure recognition data. The goal primarily is to detect the hidden mixture component different from the class labels. We investigated different perplexities and selected the one with a satisfactory low-dimensional output. At first, perplexities between 9 and 15 gave an unsatisfactory representation with "crowd points" presented in Figure (3.8). We further increased the perplexity to 250. This however contradicts the suggestion given by the authors but the output gave a clear structure. The output however fails to reveal the hidden cluster of the epileptic patients even after $10,000$ iterations [Figure (3.10)]. Afterwards, the output with the perplexity $= 250$ was filtered into the CWMs model. At this junction, we applied the 14 parsimonious models, and we observed a varying computation time due to their varying model complexities. The model selection was

performed through eight different information criteria. We observed that the number of mixture component selected BIC did not agree with ICL. While the BIC selected the models with wrong number of components, ICL selected the model EEE with the correct number of hidden components. The output is provided in Figure (3.11). However, the overlap reduced drastically when compared to Figure (3.10). The data we have used in this Chapter are categorical data with class label more than two classes. All the class labels are first transformed to be continuous variables. This is necessary because the linear Gaussian CWMs models uses OLS for the maximization step and it can only handle a continuous dependent variable efficiently. The possible feature direction should be to create a self-sufficient CWMs by embedding a dimensionality reduction technique into the **CWMs** package in **R**. This will allow the package to handle high-dimensional data. In the next chapter, we tackle the limitation of the family of CWMs and mitigate the effect of the 'curse of dimensionality' on CWMs by developing an appropriate model that is suitable for categorical data in high-dimensional space.

Chapter 4

Variational Bayesian: EM–IRLS & EM–SGD Multinomial CWM

4.1 Introduction

In order to completely combat the inability of CWMs to handle categorical data and failure in the presence of high-dimensional data, in this Chapter we develop a novel model called Multinomial Cluster Weighter Model (MCWM). MCWM is well suited for categorical data and has the capacity to handle high-dimensional data. First, MCWM allows for the possible nonlinear dependencies in the mixture components by considering a multinomial logit regression or softmax regression for multi-class. Secondly, MCWM considers multinomial distribution for the conditional distribution of the response variable given the covariates. We investigate the conditions under which the proposed model is identifiable.

The new model uses both Iteratively Reweighted Least Squares (EM-IRLS) and Stochastic Gradient Descent (EM-SGD) in the maximization step of the EM algorithm. Conventionally, maximum likelihood estimates are derived using the Expectation Maximization (EM) algorithm with OLS (EM-OLS) in the maximization step for linear Gaussian CMWs. On the contrary, we derive the EM with a Mini Batch Stochastic Gradient Descent (EM-SGD) to overcome the drawback of unscalability and matrix inversion of the model arising from the EM-OLS and EM-IRLS algorithms. Model selection is carried out using the Akaike Information Criterion (AIC),

Bayesian information criterion (BIC), and Integrated completed likelihood (ICL) and other five variants. Adjusted Rand Index variants such as Rand Index, Hubert and Arabie's (HA), Fowlkes and Mallow's (FM), Morey and Agresti's (MA), and Jaccard (JA) are considered as a different measure of accuracy. The clustering performance of the proposed model is investigated through simulated and real data sets. Considering different datasets, MCWM shows excellent clustering results via performance measures such as Accuracy and Area under the ROC curve.

4.1.1 Main contribution

The goal of this chapter is to propose an extension of a binomial CWM to a multiclass called Multinomial CWM and give an extensive derivation. We derive the identifiability condition of the proposed model. On this proposed model, we study the Expectation-Maximization algorithm from two angles such as the angle of Iteratively Reweighted Least Squares and Stochastic Gradient Descent. In particular to solve the problem of singularity of matrix inversion arising from the EM-IRLS and to make EM scalable to large dataset, we derive EM-SGD for MCWM, where we optimize EM parameter in batches. At the E step, we follow the optimization conventionally, but at the M step, we use SGD and IRLS for the multinomial distribution.

For the first time, we apply the variant of CWM to image classification problems in high-dimensional data, taking CWMs from the perspective of regression to a classification perspective. Moreover, employing the mini batch SGD, MCWM is scalable to a large dataset. Unlike the conventional EM which has to deal with the problem of matrix inversion and local maxima, EM-SGD is able to overcome these problems due to its random nature.

4.2 The Model

Multinomial Cluster Weighter Model (MCWM) is a technique that takes as a local model any form of stand alone non-linear model suitable for categorical data.
Let $(\mathbf{X}, \mathbf{Y})$ be a pair of random vector $\mathbf{X}$ and multi-class response variable $\mathbf{Y}$ defined on $\mathcal{D}$ with a joint probability $p(\mathbf{x}, \mathbf{y})$, where $\mathbf{X}$ is a d-dimensional input vector with

values in some space $\mathcal{X} \subseteq \mathrm{R}^d$ and $\mathbf{Y}$ is a J- dimensional response variable having values in $\mathcal{Y} \subseteq \mathrm{R}^J$. Thus, $\mathbf{x}, \mathbf{y} \in \mathcal{X} \times \mathcal{Y} \subseteq \mathcal{R}^{d+J}$. Suppose that $\mathcal{D}$ can be partitioned into G disjoint groups, say $\mathcal{D}_1, ..., \mathcal{D}_G$, that is $\mathcal{D} = \mathcal{D}_1 \cup ... \cup \mathcal{D}_G$. MCWM decomposes the joint probability $p(\mathbf{X}, \mathbf{Y})$ as follows:

$$p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\Theta}) = \sum_{g=1}^{G} p(\mathbf{Y}|\mathbf{X}, \mathcal{D}_g) p(\mathbf{X}|\mathcal{D}_g) \pi_g \tag{4.1}$$

where $p(\mathbf{Y}|\mathbf{X}, \mathcal{D}_g)$ is the conditional density of the multiclass response variable $\mathbf{Y}$ given the predictor vector $\mathbf{x}$ and $\mathcal{D}_g$, $p(\mathbf{x}|\mathcal{D}_g)$ is the probability density of $\mathbf{x}$ given $\mathcal{D}_g$, $\pi_g = p(\mathcal{D}_g)$ is the mixing weight of $\mathcal{D}_g$, with constraints such as $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$, $g = 1, ..., G$ and the $\boldsymbol{\Theta} = (\boldsymbol{\Omega}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)$ denotes the set of all model parameters, where $\boldsymbol{\Omega} \in \mathrm{R}^{d \times J \times G}$ denotes the coefficient of the local model, location parameter $\boldsymbol{\mu} \in \mathrm{R}^{d \times G}$, G is the number of groups, $\boldsymbol{\Sigma}$ is the positive definite covariance matrix.

In the framework of MCWM, the conditional density of multiclass response variable is assumed to be a Multinomial distribution whose probabilities are the multinomial logit regression or softmax regression, and the marginal density is taken to be a Gaussian, with $\mathbf{Y}|\mathbf{x}, \mathcal{D}_g \sim Multi(\boldsymbol{\phi}_{jg}, ..., \boldsymbol{\phi}_{Jg})$, and $\mathbf{X}|\mathcal{D}_g \sim \mathcal{N}_d(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ respectively. Thereafter, we shall write $p(\mathbf{X}|\mathcal{D}_g) = \psi_d(\mathbf{X}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ and $p(\mathbf{Y}|\mathbf{X}, \mathcal{D}_g) = p(\mathbf{y}; \boldsymbol{\phi}_g)$, $g = 1, .., G$, where the conditional densities are based on the nonlinear mappings which we define later. Thus, we get:

$$p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g p(Y_1 = y_1, ..., Y_J = y_J|\boldsymbol{\phi}_{jg}, ..., \boldsymbol{\phi}_{Jg}) \psi_d(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \tag{4.2}$$

The approach in Equation (4.2) is referred to as Multinomial CWM. In particular, the posterior is given by

$$p(\mathcal{D}_g|\mathbf{X}, \mathbf{Y}) = \frac{\pi_g p(Y_1 = y_1, ..., Y_J = y_J|\boldsymbol{\phi}_{jg}, ..., \boldsymbol{\phi}_{Jg}) \psi_d(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{k=1}^{G} \pi_k p(Y_1 = y_1, ..., Y_J = y_J|\boldsymbol{\phi}_{jk}, ..., \boldsymbol{\phi}_{Jk}) \psi_d(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \tag{4.3}$$

In Equation (4.3), $\psi(.)$ denotes the Gaussian density and the number of free parameters for MCWM is $(GJd) + G[d(d+1)/2] + Gd + (G-1)$ where

- The number of free parameters coefficients $\boldsymbol{\Omega}$ in the conditional response variable is GJd
- The number of free parameters in the covariance matrix $\boldsymbol{\Sigma}$ is $G[d(d+1)/2]$
- The number of free parameters in the mean $\boldsymbol{\mu}$ is Gd
- The number of free parameters in the mixing probability $\boldsymbol{\pi}$ is $G-1$

4.2.1 Modeling for $\mathbf{p}(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\eta}_g)$

In order to deal with discrete or categorical response variable, we assume that $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\eta}_g)$ belongs to the exponential family. Thus, in general, $\mathcal{Y} \subseteq \mathrm{R}^J$. There exists an association between exponential family and the generalized linear models via a monotone and a differentiable link function $f_g(.)$ to relate the expected value of $\boldsymbol{Y}|\boldsymbol{\phi}_j$ to the covariates $\mathbf{X}$ through the relation $\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg}) = \boldsymbol{x}_i'\boldsymbol{\beta}_{jg}$, where

$$\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg}) = \log\frac{\phi_{ijg}}{\phi_{i1g}} = \boldsymbol{x}_i'\boldsymbol{\beta}_{jg} \tag{4.4}$$

where β_{0jg} is an intercept and $\boldsymbol{\beta}_{1jg}$ is a vector of regression coefficients, for $j = 2, ..., J$. Here, we assume that the intercept β_{0jg} is implicit in Equation (4.4) and the model matrix $\mathbf{X}$ includes a column of ones.
Equation (4.4) is analogous to logistic regression model, expect that the probability distribution of the response is multinomial instead of binomial, hence Multinomial Cluster Weighted Model. Also, instead of one equation in logistic regression model, we have $J-1$ equations. We contrast each of categories $2, ..., J$ with category 1 in the multinomial logit equations, whereas in single logistic regression equation in binomial cluster weighted model [Ingrassia et al. (2015)] is a contrast between success and failure. So, if $J = 2$ the multinomial cluster weighted model reduces to binomial cluster weighted model. The multinomial logit model can also take the form of the original probabilities ϕ_{i1} rather than the log-odds. Starting from Equation (4.4), we

can write

$$\boldsymbol{\phi}_{jg} = \frac{\exp\{\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg})\}}{1+\sum_{j=2}^{J}\exp\{\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg})\}} \tag{4.5}$$

The multinomial logit model may also be written as the original probabilities ϕ_{jg} rather than the log-odds. Starting from Equation (4.5), we adopt the convention that $\phi_{1g} = 0$, then we have

$$\boldsymbol{\phi}_{jg} = \frac{\exp\{\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg})\}}{\sum_{j=1}^{J}\exp\{\boldsymbol{f}_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg})\}} \tag{4.6}$$

for $j = 1, .., J$, summing over j then $\boldsymbol{\phi}_{1g} = 1/\sum_j \exp(\boldsymbol{f}_{jg})$.

4.2.2 The Multinomial CWM

Consider a random Y_i that takes one of the several discrete values, which is indexed $1, 2, ..., J$. Let $\phi_{ij} = Pr(Y_i = j)$ denote the probability that the ith response falls in the jth category. The probability of the counts Y_{ij} given by ϕ_{ij} yields the multinomial distribution:

$$Pr(Y_{i1} = y_{ij}, ..., Y_{iJ} = y_{iJ}|\phi_{ij}, ..., \phi_{iJ}) = \binom{M_i}{y_{i1}, ..., y_{iJ}} \prod_{j=1}^{J} \phi_{ij}^{y_{ij}} \tag{4.7}$$

In Equation (4.6), one can imagine that for J possible outcomes, running $J-1$ independent binary logistic regression models where one outcome is a pivot and the other $J-1$ is regressed against the pivot outcome. Thus the posterior probability of Equation (4.3) is

$$p(\mathcal{D}_g|\mathbf{y}_i, \mathbf{x}_i) = \frac{\left(\binom{M_i}{y_{i1},...,y_{iJ}}\phi_{ijg}^{y_{ij}}...\phi_{iJg}^{y_{iJ}}\right)\mathcal{N}\left(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}\right)\pi_g}{\sum_{k=1}^{G}\left(\binom{M_i}{y_{i1},...,y_{iJ}}\phi_{ijk}^{y_{ij}}...\phi_{iJk}^{y_{iJ}}\right)\mathcal{N}\left(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k}\right)\pi_k} \tag{4.8}$$

4.2.3 Identifiability

Identifiability problems arising from the finite mixture model can be categorized into two viz; *trivial* problems and *generic* problems, [Fruhwirth-Schnatter (2006)]. Trivial identifiability is the problems of empty components, that arise as a result of components with the same parameters. Trivial problems can be avoided by restraining the feasible parameter space ω to $\tilde{\omega} \subset \omega$, $\forall\ \Theta \in \tilde{\omega}$ such that

$$\pi_g > 0 \quad g = 1, ..., G, \tag{4.9}$$

imposes a suitable ordering constraint. It has been shown that mixtures of binomial distributions with respect to generic identifiability problem are identifiable if the condition $M \geq 2G - 1$ is fulfilled, where M denotes the number of repetitions for a given individual [Teicher (1963); Blischke (1964); Titterington et al. (1985)]. The restriction is necessary and sufficient for the model class of all mixtures with a maximum of G components. The result obtained by Lindsay (1995) for the more general class of mixtures of discrete exponential densities with $M + 1$ point support and the same condition is applied for mixtures of multinomial distributions [Grun (2002); Elmore & S. (2003)]. The identifiability of mixture of Gaussian regression models is analyzed in Hennig (2000). Their results show that full rank covariance matrix is not sufficient. In addition, checking the coverage conditions is very vital in order to ensure identifiability. Supplementary results for the finite mixture of Gaussian regression models with two components, only local identifiability is considered [Meijer & Ypma (2008)]. Sufficient identifiability conditions imply that any mixture distribution function from the specified model class can be uniquely parameterized, i.e. the parameters can be uniquely determined given infinitely many observations. In contrast, if a mixture distribution is not identifiable, the parameters can still not be uniquely determined even if an infinite amount of data is available.

In order to estimate the parameters of model in Equation (4.2), it is important to establish its identifiability. Consider a parametric class of density function

$$\mathcal{F} = \{f(\boldsymbol{x}; \theta) : \boldsymbol{x} \in \mathcal{X}, \theta \in \boldsymbol{\zeta}\} \tag{4.10}$$

and the class of finite mixture of functions in $\mathcal{F}$,

$$\mathcal{H} = \Bigg\{ h(\boldsymbol{x};\boldsymbol{\varphi}) : h(\boldsymbol{x};\boldsymbol{\zeta}) = \sum_{g=1}^{G} f(\boldsymbol{x};\boldsymbol{\zeta}_g)\pi_g, \quad \text{with} \quad \pi_g > 0 \quad \text{and} \quad \sum_{g=1}^{G}\pi_g = 1,$$

$$f(.;\theta_g) \in \mathcal{F}, g = 1, ..., G, \boldsymbol{\zeta}_g \neq \boldsymbol{\zeta}_k \text{ for } g \neq k, G \in \mathcal{N}, \boldsymbol{x} \in \mathcal{X}, \boldsymbol{\varphi} \in \boldsymbol{\Theta} \Bigg\} \tag{4.11}$$

This class is identifiable, if for any two members of $\mathcal{H}$ such that

$$h(\boldsymbol{x};\boldsymbol{\varphi}) = \sum_{g=1}^{G} f(\boldsymbol{x};\boldsymbol{\theta}_g)\pi_g, \quad \text{and} \quad h(\boldsymbol{x};\tilde{\boldsymbol{\varphi}}) = \sum_{v=1}^{\tilde{G}} f(\boldsymbol{x};\tilde{\boldsymbol{\theta}}_v)\tilde{\pi}_v \tag{4.12}$$

the equality $h(\boldsymbol{x};\boldsymbol{\varphi}) = h(\boldsymbol{x};\tilde{\boldsymbol{\varphi}})$ implies that $G = \tilde{G}$ and there exists a one-to-one correspondence between the two sets $\{1, ..., G\}$ and $\{1, ..., \tilde{G}\}$, such that $\pi_g = \tilde{\pi}_v$ and $\theta_g = \tilde{\theta}_v$. Here, we wish to establish the identifiability of MCWM defined in Equation 4.2. The class of the MCWM is given by

$$\mathcal{P} = \Bigg\{ p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\varphi}) : p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\varphi}) = \sum_{g=1}^{G} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\beta}_{jg})\psi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g,$$

$$\text{with} \quad \pi_g > 0, \sum_{g=1}^{G}\pi_g = 1, \boldsymbol{\beta}_j^g \neq \boldsymbol{\beta}_j^s$$

$$\text{for} \quad g \neq s, (\boldsymbol{x}', \boldsymbol{y})' \in \mathcal{R} \times \mathcal{Y}, \boldsymbol{\varphi} = \{\boldsymbol{\beta}_j^g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \pi_g; g = 1, ..., G\} \in \boldsymbol{\Theta}, G \in \mathcal{N} \Bigg\} \tag{4.13}$$

where $\mathcal{Y}$ depends on the component distribution q. We provide the sufficient conditions for $\mathcal{P}$ to be identifiable in $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathcal{R}^d$ is a set with probability one according to the $d-$variate Gaussian density ϕ.

Theorem

Let $\mathcal{P}$ be the class defined in Equation (4.7) and assume that there exists a set $\mathcal{X} \subseteq \mathcal{R}^d$ with probability one of Gaussian density such that the mixture of multinomial distributions

$$\sum_{g=1}^{G} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}^g)\gamma_g(\boldsymbol{x}) \tag{4.14}$$

where

$$\ln\left[\frac{\boldsymbol{\theta}_j^g}{\boldsymbol{\theta}_J^g}\right] = \boldsymbol{x}_i'\boldsymbol{\beta}_j^g \tag{4.15}$$

is identifiable for each fixed $\boldsymbol{x} \in \mathcal{X}$*, where* $\gamma_1(\boldsymbol{x}), ..., \gamma_G(\boldsymbol{x})$ *are positive weights summing to one for each* $\boldsymbol{x} \in \mathcal{X}$*. Then the class* $\mathcal{P}$ *is identifiable in* $\mathcal{X} \times \mathcal{Y}$*, if the following conditions are fulfilled:*
For all $j = 1, ..., J-1$ *there exists a non-empty* $\tilde{\boldsymbol{I}}_g$ *which is a subset of* $\cup_i \boldsymbol{I}_i$ *and for which*

$$\sum_i \sum_j M_{ij} \geq 2G - 1 \ \ \forall \ \ i \in \tilde{\boldsymbol{I}}_g. \tag{4.16}$$

where $\tilde{\boldsymbol{I}}_g$ *is defined as the index sex of all observation for the individual* i *with covariate vector* $\boldsymbol{x}_i$*.* The condition guarantees that no intra-component label switching is possible. Intra-component label switching is an identifiability problem where the labels fixed in one covariate point according some ordering constraints, the labels may switch in order covariate points for the different parameterizations of the model. As the component membership is fixed for each individual, there exists a hyperplane that separates the components and the only feasible hyperplanes are those that partition the covariate points where the covariate points from the same individual fall on the same side of the hyperplanes. The condition implies that there exists a $i \in N$ with at least $2G - 1$ observations. The proof of this theorem is deffered to Appendix (B).

4.3 The EM-IRLS and EM-SGD Algorithms for Parameter Estimation

Let $(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_N, \boldsymbol{y}_N)$ be a sample from drawn from model in Equation (4.2). The corresponding likelihood, for a fixed number of components G, is given by

$$L(\boldsymbol{\Theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i, \boldsymbol{y}_i; \Big|\boldsymbol{\Theta}) = \prod_{i=1}^{N}\sum_{g=1}^{G} \pi_g p(\boldsymbol{y}_i|\boldsymbol{\phi}_{ig}) \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}) \tag{4.17}$$

Define $z_i = (z_{i1}, ..., z_{iG})'$, with $z_{ig} = 1$ if $(\boldsymbol{x}_i', \boldsymbol{y}_i')'$ comes from $\mathcal{D}_g$, and $z_{ig} = 0$ otherwise, and consider the complete data $\left\{\left(\boldsymbol{x}_i', \boldsymbol{y}_i', \boldsymbol{z}_i'\right)'; i = 1, ..., N\right\}$. Then the complete-data

likelihood is as follows;

$$L_{\mathcal{C}}(\boldsymbol{\Theta}) = \prod_{i=1}^{N}\prod_{g=1}^{G}\pi_g^{z_{ig}} p(\boldsymbol{y}_i|\boldsymbol{\phi}_{ig})^{z_{ig}} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g})^{z_{ig}} \tag{4.18}$$

where z_{ig} denotes the gth component of $\mathbf{z}_i$. Taking the logarithm of Equation (4.18), we obtain

$$l_{\mathcal{C}}(\boldsymbol{\Theta}) = \sum_{i=1}^{N}\sum_{g=1}^{G}\Bigg[z_{ig}\ln p(\boldsymbol{y}_i|\boldsymbol{\phi}_{ig}) + z_{ig}\Big\{\ln\pi_g + \ln\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g})\Big\}\Bigg] \tag{4.19}$$

To maximize Equation (4.19) is the main goal of mixture model. Now, to prepare Equation (4.19) for the optimization technique, we write

$$\sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln p(\mathbf{y}_i|\boldsymbol{\phi}_{ig}) + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\Big\{\ln\pi_g + \ln\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g})\Big\} \tag{4.20}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\beta}_{1g}, ..., \boldsymbol{\beta}_{Jg}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \pi_g)$ are the parameters to estimate.
From Equation (4.19), we have

$$l_{\mathcal{C}}(\boldsymbol{\Theta}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\left\{\prod_{j=1}^{J}\phi_{ijg}^{y_{ij}}\right\} + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}) + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\pi_g \tag{4.21}$$

since z_i is a vector of ones and zeros, we can also write Equation (4.21) as

$$\sum_{i=1}^{N}\sum_{g=1}^{G} 1\{z_i = g\}\ln\left\{\prod_{j=1}^{J}\phi_{ijg}^{y_{ij}}\right\} + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}) + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\pi_g \tag{4.22}$$

where $1\{z_i = g\}$ is a vector of ones only where g is true. Equation 4.21 can be rewritten as

$$l_c(\boldsymbol{\Theta}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\left\{\prod_{j=1}^{J}\phi_{ijg}^{y_{ij}}\right\} + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}) + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\ln\pi_g \tag{4.23}$$

where $\boldsymbol{\Omega}' = (\boldsymbol{\beta}'_{01g}, ..., \boldsymbol{\beta}'_{1Jg})'$, $\boldsymbol{k} = (\boldsymbol{k}'_1, ..., \boldsymbol{k}'_G)'$ with $\boldsymbol{k}_g = (\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)'$. Z is a matrix whose rows are vectors of ones and zeroes, one at the position of the group and zero

everywhere else.

$$l_{1c}(\mathbf{\Omega}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \ln \left\{ \prod_{j=1}^{J} \phi_{ijg}^{y_{ij}} \right\} \tag{4.24}$$

By using Equation 4.6, Equation (4.24) becomes

$$l_{1c}(\mathbf{\Omega}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \ln \left\{ \prod_{j=1}^{J} \left[\frac{\exp(\boldsymbol{\beta}_{0jg} + \boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})}{\sum_{j}^{J} \exp(\boldsymbol{\beta}_{0jg} + \boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})} \right]^{y_{ij}} \right\} \tag{4.25}$$

$$l_{1c}(\mathbf{\Omega}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \left\{ \sum_{j=1}^{J} y_{ij} \ln \left[\frac{\exp(\boldsymbol{\beta}_{0jg} + \boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})}{\sum_{j}^{J} \exp(\boldsymbol{\beta}_{0jg} + \boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})} \right] \right\} \tag{4.26}$$

$$l_{2c}(\boldsymbol{k}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \ln \psi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{4.27}$$

$$l_{3c}(\boldsymbol{\pi}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \ln \pi_g \tag{4.28}$$

4.3.1 E-step

The EM algorithm [Dempster et al. (1977)] can be used to maximize $l_{\mathcal{C}}(\mathbf{\Omega})$, $l_{\mathcal{C}}(\boldsymbol{k})$ and $l_{\mathcal{C}}(\boldsymbol{\pi})$ to find the maximum likelihood (ML) estimates for the unknown parameters of the MCWM. The E and M steps of the algorithm can be detailed as follows: The E-step, on the *qth* iteration, requires the calculation of

$$Q(\tau; \tau^{(q)}) = E_{\tau^{(q)}}[l_c(\tau)|\mathbf{\Theta}] \tag{4.29}$$

The E-step on the qth iteration simply requires the calculation of the current conditional expectation of Z_{ig} given the observed sample due to the linearity of $l_c(\tau)$ in the unobserved data z_{ig}, where Z_{ig} is the random variable of z_{ig}. Then the conditional

expectation of Z_{ig} is as follows:

$$E_{\tau^{(q)}}(Z_{ig}|\boldsymbol{\Theta}) = z_{ig}^{(q)} = \frac{\left(\binom{M_i}{y_{i1},\ldots,y_{iJ}}(\phi^{(q)})_{ijg}^{y_{ij}}\ldots(\phi^{(q)})_{iJg}^{y_{iJ}}\right)\mathcal{N}\left(\mathbf{x}_i|\boldsymbol{\mu}_g^{(q)},\boldsymbol{\Sigma}_g^{(q)}\right)\pi_g^{(q)}}{\sum_{k=1}^{G}\left(\binom{M_i}{y_{i1},\ldots,y_{iJ}}(\phi^{(q)})_{ijk}^{y_{ij}}\ldots(\phi^{(q)})_{iJk}^{y_{iJ}}\right)\mathcal{N}\left(\mathbf{x}_i|\boldsymbol{\mu}_k^{(q)},\boldsymbol{\Sigma}_k^{(q)}\right)\pi_g^{(q)}} \quad (4.30)$$

which correspond to the posterior probability that the unobserved data $(x_i, y_i)'$ belong to the gth component of the mixture, using the current fit $\boldsymbol{\tau}^{(q)}$ for $\boldsymbol{\tau}$. Substituting the values z_{ig} in Equation (4.22) with the values $z_{ig}^{(q)}$ obtained in equation (4.30), we have

$$Q(\boldsymbol{\tau};\boldsymbol{\tau}^{(q)}) = Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(q)}) + Q_2(\boldsymbol{k};\boldsymbol{\tau}^{(q)}) + Q_3(\boldsymbol{\pi};\boldsymbol{\tau}^{(q)}) \quad (4.31)$$

where

$$Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(q)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)}\left\{\sum_{j=1}^{J} y_{ij}\ln\left[\frac{\exp(\boldsymbol{\beta}_{0jg}+\boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})}{\sum_{j=1}^{J}\exp(\boldsymbol{\beta}_{0jg}+\boldsymbol{x}_i'\boldsymbol{\beta}_{1jg})}\right]\right\} \quad (4.32)$$

$$Q_2(\boldsymbol{k};\boldsymbol{\tau}^{(q)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)}\ln\psi_d(\boldsymbol{x}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g) \quad (4.33)$$

$$Q_3(\boldsymbol{\pi};\boldsymbol{\tau}^{(q)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)}\ln\pi_g \quad (4.34)$$

4.3.2 M-step

On the M-step, at the $(q+1)th$ iteration, it follows that $\boldsymbol{\Omega}^{(q)}$, $\boldsymbol{k}^{(q)}$ and $\boldsymbol{\pi}^{(q)}$ in Equation (4.21) can be computed independently of each other, by separate maximization of Equations (4.26), (4.27) and (4.28), respectively. Moreover, $\boldsymbol{k}^{(q)}$ and $\boldsymbol{\pi}^{(q)}$ can be computed in closed form but $\boldsymbol{\Omega}^{(q)}$ cannot be computed in closed form.

4.3.3 Maximization of $Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(q)})$ via IRLS

The updated estimates $\tau^{(q+1)}$ are the solutions of the following M-step.

$$\frac{\partial}{\partial\boldsymbol{\beta}}Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(q)}) = \frac{\partial}{\partial\boldsymbol{\beta}}\sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)}\left\{y_{i1}\ln\phi_{i1g} + \sum_{j=2}^{J} y_{ij}\ln\phi_{ijg}\right\} \quad (4.35)$$

Since the derivative in Equation (4.35) does not have a closed form, we resolve to iterative optimization which will be derived further in the Appendix (C). The updated estimates are

$$\beta_{jg}^{(q+1)} = \beta_{jg}^{(q)} + \left(\sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i' v_{ijg} \boldsymbol{x}_i\right)^{-1} \left(\sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i v_{ijg} \zeta_{ij}^{*}\right) \tag{4.36}$$

$$\beta_{jg}^{(q+1)} = \left(\sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i' v_{ijg} \boldsymbol{x}_i\right)^{-1} \left(\sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i v_{ijg} \zeta_{ij}^{(q)}\right) \tag{4.37}$$

where $v_{ijg} = \phi_{ijg}^{(q)}(1 - \phi_{ijg}^{(q)})$, $\zeta_{ij}^{(q)} = n_i \boldsymbol{x}_i \beta_{jg}^{(q)} + \zeta_{ij}^{*}$ and $\zeta_{ij}^{*} = y_{ij}/\phi_{ijg}^{(q)} - y_{i1}/\phi_{i1g}^{(q)}$. The weight v_{ijg} and the adjusted response $\zeta_{ij}^{(q)}$ are updated at each iteration based on the current estimates of the multinomial distribution probability ϕ_{ijg}.

4.3.4 Minimizing Negative of $Q_1(\boldsymbol{\Omega}; \boldsymbol{\tau}^{(q)})$ by SGD

MCWM wishes to explain the data by minimizing the negative log-likelihood function.

$$Q_1(\boldsymbol{\Omega}; \boldsymbol{\tau}^{(q)}) = -\sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)} \ln(\phi_{ijg}) \tag{4.38}$$

where the ϕ_{ijg} is given in Equation (4.6). The cost function in Equation (4.38) cannot be solved analytically, so we seek to minimize the cost function by using the iterative optimization algorithm called Gradient descent. The free parameters to be adapted by SGD are the $\boldsymbol{\Omega} = \{\boldsymbol{\beta}_{0j1}, ..., \boldsymbol{\beta}_{0jG}\}$, which are the parameters of the conditional response variable. The derivative is as follows;

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = -\frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)} \ln(\phi_{ijg}) \tag{4.39}$$

using the chain rule and following from Equation (4.4) and $f(\boldsymbol{x}, \boldsymbol{\beta}_g) = \boldsymbol{\beta}_{0jg} + \boldsymbol{x}'\boldsymbol{\beta}_{1jg}$, the gradient of Equation (4.39) can be written and derived as follows

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = \frac{\partial}{\partial \boldsymbol{\phi}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) \frac{\partial}{\partial f(\boldsymbol{x}; \boldsymbol{\beta})} \boldsymbol{\phi}(f) \frac{\partial}{\partial \boldsymbol{\beta}} f(\boldsymbol{x}; \boldsymbol{\beta}) \tag{4.40}$$

Then,

$$\frac{\partial}{\partial \phi_{ig}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)} \frac{1}{\phi_{ijg}} \tag{4.41}$$

The next equation will be derived on element-by-element basis, that is;

$$\frac{\partial}{\partial f_{ijg}(\boldsymbol{x};\boldsymbol{\beta})} \boldsymbol{\phi}(f_{jg}) = \frac{\partial}{\partial f_{jg}(\boldsymbol{x};\boldsymbol{\beta})} \frac{\exp\{f_{jg}(\boldsymbol{x};\boldsymbol{\beta}_{jg})\}}{\sum_{g=1}^{G}\exp\{f_k(\boldsymbol{x};\boldsymbol{\beta}_k)\}} \tag{4.42}$$

For $i = j$: Equation (4.42) is $\phi_{ijg}(1-\phi_{ijg})$ and for $i \neq j$, Equation (4.42) is $-\phi_{ijg}\phi_{ijg}$. The gradients of the intercept and the coefficients are given as follows

$$\beta_{0jg}^{(q+1)} = \beta_{0jg}^{(q)} - \alpha \frac{\partial}{\partial \beta_{0jg}} Q_1(\boldsymbol{\beta}; \boldsymbol{\psi}^{(q)}) \tag{4.43}$$

$$\boldsymbol{\beta}_{1jg}^{(q+1)} = \boldsymbol{\beta}_{1jg}^{(q)} - \alpha \frac{\partial}{\partial \boldsymbol{\beta}_{1jg}} Q_1(\boldsymbol{\beta}; \boldsymbol{\psi}^{(q)}) \tag{4.44}$$

A learning rate α is required for performing the SGD. We observe that a good choice of the learning rate for MCWM which can either be fixed or tuned in the algorithm is between 0.01 and 0.09. Generally, we can always set the batch size to 1. However, the limitation of this is the longer time it takes to reach convergence. Therefore for high-dimensional data, increasing the batch size is required.

4.3.5 Maximizing $Q_2(\boldsymbol{k}; \boldsymbol{\tau}^{(q)})$ by ML

With reference to the updated estimates of $\boldsymbol{k}_g$, $g = 1, ..., G$,

$$Q_2(\boldsymbol{k}; \boldsymbol{\tau}^{(q)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ijg}^{(q)} \ln \psi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{4.45}$$

maximizing Equation (4.45) with respect to $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$, $g = 1, ..., G$ is equivalent to independently maximizing each of the G expressions that leads to the following results;

$$\sum_{i=1}^{N} z_{ig}^{(q)} \ln \psi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{4.46}$$

$$\boldsymbol{\mu}_g^{(q+1)} = \sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i \Big/ \sum_{i=1}^{N} z_{ig}^{(q)} \tag{4.47}$$

and the covariance matrix for the component is as follows

$$\boldsymbol{\Sigma}_g^{(q+1)} = \sum_{i=1}^{N} z_{ig}^{(q)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_g^{(q+1)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_g^{(q+1)}\right)^T \Big/ \sum_{i=1}^{N} z_{ig}^{(q)} \tag{4.48}$$

4.3.6 Maximizing $Q_3(\boldsymbol{\pi}; \boldsymbol{\tau}^{(q)})$ by ML

Regarding the mixing weights, maximization of $Q_3(\boldsymbol{\pi}; \boldsymbol{\tau}^{(q)})$ with respect to $\boldsymbol{\pi}$ subject to the constraints on those parameters is obtained by maximizing the Lagrangian function

$$\sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)} \ln \pi_g - \lambda\left(\sum_{g=1}^{G} \pi_g - 1\right) \tag{4.49}$$

where λ is the Lagrangian multiplier. Setting the derivative of Equation (4.49) with respect to π_g to zero and solving for π_g yields the solution

$$\pi_g^{(q+1)} = \frac{1}{N} \sum_{i=1}^{N} z_{ig}^{(q)} \tag{4.50}$$

4.3.7 Algorithm for MCWM

Algorithm 3 Algorithm for Multinomial Cluster-Weighted Models

Insert training Data $(\mathbf{X}, \mathbf{Y})$
select an initial coefficient $\boldsymbol{\beta}$;
initialize the group mean and Covariance matrix $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively,
initialize assignment probability π_g
while $q \neq$ maxit **do**
  while $g \neq$ G **do**
    while $j \neq$ J **do**
      Compute $f(\mathbf{x}_i; \boldsymbol{\beta}_g) = \boldsymbol{x}_i' \boldsymbol{\beta}_{jg}$
      Compute the $\boldsymbol{\phi}_{jg}$ using Equation (4.5)
      E-Step: Compute the posterior probability z_{ig}; $p(z_{ig}|\boldsymbol{x}_i)$ using Equation (4.30)
      M-Step: Update the parameters as follows:
      compute $\boldsymbol{\mu}^{(q+1)} \leftarrow \sum_{i=1}^{N} \boldsymbol{x}_i z_{ig}^{(q)} \Big/ \sum_{i=1}^{N} z_{ig}^{(q)}$
      compute $\boldsymbol{\Sigma}^{(q+1)} \leftarrow \sum_{i=1}^{N} z_{ig}^{(q)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_g^{(q+1)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_g^{(q+1)}\right)^T \Big/ \sum_{i=1}^{N} z_{ig}^{(q)}$
      compute $\boldsymbol{\pi}^{(q+1)} \leftarrow \sum_{i=1}^{N} z_{ig}^{(q)} \Big/ N$
      compute the update of $\beta_{jg}^{(q+1)}$ according to Equation (4.36) and Equation (4.37) for EM-IRLS or Equation (4.43) and Equation (4.44) for EM-SGD
      compute the complete log-likelihood function $l_c(\boldsymbol{\Theta})$ using Equation (4.23)
    end while
  end while
end while
return $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Omega}, \boldsymbol{\pi}, l_{\mathcal{C}}(\boldsymbol{\Theta}), \boldsymbol{Z}$

4.4 Computational Issues

Codes for the of EM algorithm described in section (4.3) was written in **R** computing environment [R (2019)]. EM algorithm is an iterative, strictly hill-climbing whose performance or behavior can be hampered or determined by the choice of the starting values. EM algorithm is very sensitive to its starting values. On the other hand, the starting values may cause a numerical instability or explosion due to singularity.

This sensitivity to starting values is inevitable because the likelihood function often has multiple local maxima [McLachlan & Peel (2000)]. Thus, to achieve the best of the EM algorithm, good initialization is crucial for finding ML estimations. Many suggestions about the selection of initial values have been provided in the literature [e.g. Figueiredo & Jain (2002); Maitra (2009)]. However, no strategy is superior to the other. The use of hierarchical clustering was proposed by Banfield & Raftery (1993) and incorporated in **R** package **Mclust** for Gaussian mixture. It works well with well-separated or less overlapping components. This however, may be infeasible for initialization when clustering large data sets. There are also stochastic algorithms for initialization such as *em*EM algorithm proposed by Biernacki et al. (2003) which consists of two parallel EM runs. The first stage, called short *em*, involves starting from several random values and use the result as a starting value for the EM until convergence is reached. The advantage gained from this strategy of initialization is a fast convergence. However, it does not guarantee that good estimates for component will be found. As a modification to *em*EM, Maitra (2009) replaced the short *em* with a *Rnd* by choosing multiple starting points and evaluating log-likelihood at these values without running any EM iterations. However, Baudry et al. (2010) argued that using multiple random starting point in quest of global maximum can be time-consuming, such standard initialization consists in selecting a value for $\tau^{(0)}$. An alternative approach is to perform the first E-step by specifying in the equation, the values of $\boldsymbol{z}_i^{(0)}$, $i = 1, ..., N$ [Forina (1991); McLachlan & Peel (2000), p. 54].

4.4.1 Convergence Criterion

To monitor convergence, the stopping criterion is usually adopted with the EM algorithm in terms of either the relative small deflection in the parameter estimates or the log-likelihood, $\log(L(\boldsymbol{\Theta}))$. However, as Lindstrom & Bates (1991) emphasize, this is just a measure of lack of movement between current log-likelihood and the previous log-likelihood but not of actual convergence. As established by Bohning et al. (1994), in their application to the sequence of the log-likelihood values to provide a useful

estimate of the limiting values, Aitken's acceleration is applicable in the case where the sequence of the log-likelihood values $l^{(q)}$ is linearly convergent to some l^*.

We adopted the Aitken's acceleration method to monitor the convergence of the EM-IRLS algorithm. Here the qth iteration of the log-likelihood is

$$l^{(q)} = \log L(\boldsymbol{\Theta}^{(q)}) \tag{4.51}$$

Under this assumption,

$$l^{(q+1)} - l^* \approx a(l^{(q)} - l^*), \tag{4.52}$$

for all q and some $a \in (0, 1)$, a decision can be made based on this estimates whether or not the algorithm has reached convergence that is, whether or not the log-likelihood is close to the estimated asymptotic value. The Aitken acceleration at iteration q is given by

$$a^{(q)} = \frac{l^{(q+1)} - l^{(q)}}{l^{(q)} - l^{(q-1)}} \tag{4.53}$$

where $l^{(q+1)}, l^{(q)}$, and $l^{(q-1)}$ are the log-likelihood values from iterations $q+1, q$ and $q-1$, respectively. Then, the asymptotic estimate of the likelihood at iteration $q+1$ is given by

$$l_\infty^{(q+1)} = l^{(q)} + \frac{1}{1 - a^{(q)}}(l^{(q+1)} - l^{(q)}). \tag{4.54}$$

In a situation where the primary interest is on the sequence of the log-likelihood values rather than the sequence of the parameter estimates, Bohning et al. (1994) suggest the EM algorithm can be stopped if $\left|l_\infty^{(q+1)} - l^{(q)}\right| < \epsilon$ for small ϵ. Following McNicholas (2010) and (n.d.), we stopped the algorithm with $\left|l_\infty^{(q+1)} - l^{(q)}\right| <= 0.05$.

4.4.2 Model selection and performance evaluation

In mixture models, it is a common practice to assume that the functional form and the variables of the mixing densities are known. However, in the past, model selection has typically been referred to the problem of choosing the optimal number of components G. Moreover, in the recent investigations, identification of variables with more predictive power has been carried out.

4.4.3 Receiver's Operating Characteristics Curve

A receiver's operating characteristics (ROC) graph is a visualizing, organizing and classifying technique based on their performance. ROC grapghs have been used for many detection theories to depict the tradeoff between hits rates and false alarm rates of classifiers [Egan (1975); Swets et al. (2000)]. To distinguish between the actual class and the predicted class we use the labels $pred, real$ for the class prediction produced by MCWM. Given a classifier and an instance, there are four possible outcomes; if the instance is positive and it is correctly classified as positive then it is a *true positive*; if it is classified as negative it becomes the *false negative*. Similarly, if an instance is negative and the classifier correctly classifies it as negative then it is a *true negative* but if incorrectly classified it is a *false positive*. In the confusion matrix, the diagonal values represents the correctly classified labels and the number off this diagonal represents the error or confusion between various classes. The true positive rate (also called *hit* and *recall*) of a classifier is estimated as

$$\text{tp rate} \approx \frac{\text{Positive correctly classified}}{\text{Total positives}} \tag{4.55}$$

$$\text{fp rate} \approx \frac{\text{Negatives incorrectly classfied}}{\text{Total Negatives}} \tag{4.56}$$

Additional terms are associated with ROC curves are $sensitivity = recall$, while

$$\begin{aligned} \text{specificity} &= \frac{\text{True negative}}{\text{False positives} + \text{True negatives}} \\ &= 1 - \text{fp rate} \end{aligned} \tag{4.57}$$

positive predictive value = precision.

4.5 A Simulation Study for Multinomial CWM

A simulation study was performed to evaluate the performance of the MCWM obtained via EM algorithm. The data were simulated from Multinomial CWM according to the Equation (4.8) above. We considered two different scenarios of groups $G = 2$ and $G = 3$ with two sample sizes of $n = 500$ and $n = 1000$ in the simulation. First,

we chose values for the parameters; the number of covariates $d = 2$, the response variable Y is a categorical variable with 3 levels a, b, c where c is the baseline for both a and b.

Among the possible initialization strategies proposed, we adopted the random starting values to estimate multinomial cluster weighted model. The choice is due to its simplicity compared to other proposed initialization methods. However, to ensure a near-optimal likelihood value, we repeated the algorithm more than once and select the solution with the highest value of likelihood. The algorithm may explode due to a bad random start-off which may result into singularity. Some issues arising from the random stating values can also lead to non-convergence in the IRLS algorithm. To alleviate this problem, we adopted more powerful and stable optimization package to execute the IRLS algorithm in the M-step. We used the **R** function **Optim_sa** from **optimization** package [Kirkpatrick et al. (1983); Pronzato et al. (1984); and Corana et al. (1987)]. The function searches the global optimum with systematic component and allows for non-linear, non-differentiable, and multimodal functions. The advantage of this is that the algorithm has the ability to escape the trap of local maximum.

4.5.1 Continuous Covariates and Mixing Proportions

We considered the two- and three-dimensional observations obtained by generating samples from each of the two multivariate Gaussian distributions. The covariance matrices are identical matrices and the means are chosen for both scenarios with $G = 2$ and $G = 3$. The vectors of means and mixing proportions for both scenarios are presented in Table (4.1 top) and Table (4.1 bottom)

Table 4.1: True values of $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and $\boldsymbol{\pi}$ for $G = 2$ (top) and $G = 3$ (bottom), $n = 500, 1000$

g	μ_1	μ_2	σ^2_{11}	σ^2_{22}	$\boldsymbol{\pi}$
1	0.10	2.00	1.00	1.00	0.50
2	−2.00	0.00	1.00	1.00	0.50
1	0.10	0.00	1.00	1.00	1/3
2	−2.00	1.00	1.00	1.00	1/3
3	2.00	3.00	1.00	1.00	1/3

4.5.2 Response Variables

The response variable Y is defined as a multinomial variable with 3 levels ($a, b,$ and c). Y is obtained by applying Equation (4.5). Fixing the intercept at zero and the slopes $\boldsymbol{\beta}$ are presented in Table (4.2 top) for $G = 2$ and Table (4.2 bottom) for $G = 3$. We tried as many values to increase the level of overlap in the observations. The presence of overlap ensures that MCWM can handle not just well separated clusters but well cluttered observations.

Table 4.2: True values of coefficients $\boldsymbol{\beta}$ for $n = 500, 1000$ and $G = 2, 3$

g	Y	β_0	β_1	β_2
1	a	0.000	5.000	0.400
	b	0.000	0.300	0.040
2	a	0.000	0.010	0.020
	b	0.000	2.000	1.000
1	a	0.000	5.000	0.400
	b	0.000	0.300	0.040
2	a	0.000	0.010	0.020
	b	0.000	2.000	1.000
3	a	0.000	1.000	0.030
	b	0.000	0.060	0.020

4.5.3 Algorithm for simulating from MCWM

Algorithm (4) is explained as follows; each data point was generated according to the following setup: first, we generated random number of length n from a uniform distribution that is U with $(0, 1)$ and the generated value was used to select a particular component from MCWM.

In line 1, we initialized the coefficients $\boldsymbol{\beta}$ according to Table (4.2). In line 2, the component mean $\boldsymbol{\mu}$, the number of mixture components G, and $\boldsymbol{\Sigma}$ were initialized as presented in Table (4.1). We generated $\boldsymbol{x}_i$ from Gaussian distribution with their respective group parameters $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$. The probability ϕ_{i1} was computed using Equation (4.5). Based on the probabilities $\boldsymbol{\phi}$, we generated response variable Y from

multinomial distribution as described in the introduction of the simulation study in Section (4.5). The algorithm is presented in Algorithm (4).

Algorithm 4 Algorithm for simulation Multinomial Cluster-Weighted Models with multinomial response variables.

Select an initial coefficient $\boldsymbol{\beta}$, Number of Groups G
Initialize the group mean and Covariance matrix $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively,
Initialize assignment probability π_g
Set a seed
Generate $U \sim (0, 1)$
while $i \neq n$ **do**
  if $U_i < \pi_1$ **then**
    Generate $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$
    Compute $\boldsymbol{\phi}_{i1}$ using equation (4.5)
    Generate $\boldsymbol{y}_i \sim Multi(1, \boldsymbol{\phi}_{i1})$
  else if $U_i > \pi_1$ & $U_i < \pi_1 + \pi_2$ **then**
    Generate $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
    Compute $\boldsymbol{\phi}_{i2}$ using equation (4.5)
    Generate $\boldsymbol{y}_i \sim Multi(1, \boldsymbol{\phi}_{i2})$
  else
    Generate $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$
    Compute $\boldsymbol{\phi}_{i3}$ using equation (4.5)
    Generate $\boldsymbol{y}_j \sim Multi(1, \boldsymbol{\phi}_{i3})$
  end if
end while

We present the estimates of the coefficients for two-component MCWM with $n = 500$ and $n = 1000$ with c as the baseline. We select eight core information criteria for selecting the true mixture components G such as presented in Table (2.2). We also evaluate the performance of the MCWM with the Receiver's operating characteristics(ROC) plot and provide the Area under ROC curve for both scenarios. Additionally, we provide the Adjusted Rand Index and its variants such as Rand Index (RI), Hubert and Arabie's adjusted Rand index, Morey and Agresti's adjusted Rand index, Fowlkes and Mallow's adjusted Rand index, and Jaccard index, which measure the agreement between the true cluster and classification result of the proposed model.

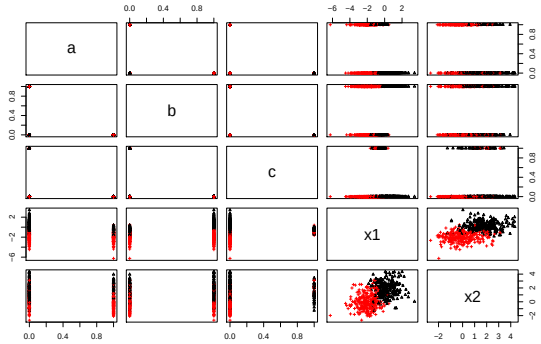


Figure 4.1: The classification plot of Multinomial CWMs for ($n = 500$, $G = k = 2$) with two covariates.

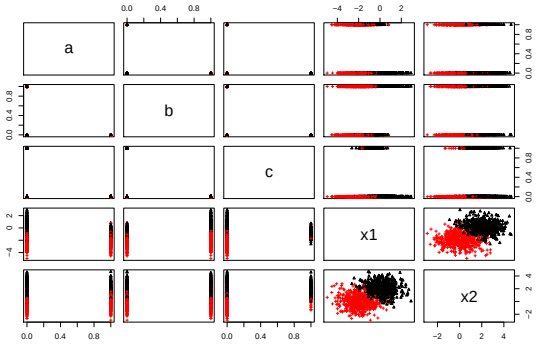


Figure 4.2: The classification plot of multinomial CWM for ($n = 1000$, $G = k = 2$) with two covariates.

4.5.4 Results for the two components

Here, we present the results produced by the MCWM when the true cluster $G = 2$. First, we pretended as if the number of components is unknown. Then we allow MCWM to discover the hidden components of the observation to ascertain the model selection power of the proposed model.

Figure (4.1) and Figure (4.2) show the plot partitioned by the classification result of MCWM for $n = 500$ and $n = 1000$ with the number of mixture component $G = 2$. Table (4.3) shows the estimates of $\boldsymbol{\mu}$, $\boldsymbol{\pi}$ and the diagonal of $\boldsymbol{\Sigma}$. When $n = 500$, the estimates provided in group 2 e.g, for $\hat{\mu}_1$ is -1.944 and for $n = 1000$, $\hat{\mu}_1$ is -2.004. The estimates for the coefficients are shown in Table (4.4) with c as the baseline.

In Table (4.5 Top), the model achieves an accuracy of 93% when $n = 500$. In group one, the model has a misclassification rate of 5.60%, while in group two the model has a misclassification rate of 8.21%. The overall misclassification rate is 7.00%. By contrast, Table (4.5 Bottom) shows the accuracy of the model with the sample size $n = 1000$ to be 93.10%. However, the misclassification rate in group two is 7.20%, while the overall misclassification rate is 6.90%. The visualization of the confusion matrices for $n = 500$ and $n = 1000$ are presented in Figure (4.3) and Figure (4.5) respectively.

Table 4.3: Estimated values of $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$ and $\boldsymbol{\pi}$ for $n = 500, 1000$ and $G = 2$

	n	g	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}^2_{11}$	$\hat{\sigma}^2_{22}$	$\hat{\boldsymbol{\pi}}$
True		1	0.100	2.000	1.000	1.000	0.500
		2	−2.000	0.000	1.000	1.000	0.500
Recovered	500	1	0.193	1.821	0.968	1.159	0.453
		2	−1.944	0.046	0.946	1.245	0.547
Recovered	1000	1	0.098	1.997	0.921	0.974	0.484
		2	−2.004	−0.023	0.912	1.058	0.516

Table 4.4: Estimated values of coefficients $\boldsymbol{\beta}$ for $n = 500, 1000$ and $G = 2$ with c as the baseline

	n	g	Y	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
True		1	a	0.000	5.000	0.400
			b	0.000	0.300	0.040
		2	a	0.000	0.010	0.020
			b	0.000	2.000	1.000
Recovered	500	1	a	0.298	6.889	0.159
			b	0.593	0.377	−0.489
		2	a	0.348	0.143	0.156
			b	0.602	3.793	2.597
Recovered	1000	1	a	0.378	4.887	0.183
			b	−0.118	0.160	−0.037
		2	a	0.117	−0.004	−0.048
			b	−0.133	1.721	0.608

Table 4.5: Confusion Matrix in the three component Model for $n = 500$ and 1000

n	Component	1	2	MR (%)
500	1	219	13	5.60
	2	22	246	8.21
Misclassification	7.00			
Accuracy	93.00%			
1000	1	451	35	7.20
	2	34	480	6.61
Misclassification	6.90			
Accuracy	93.10%			

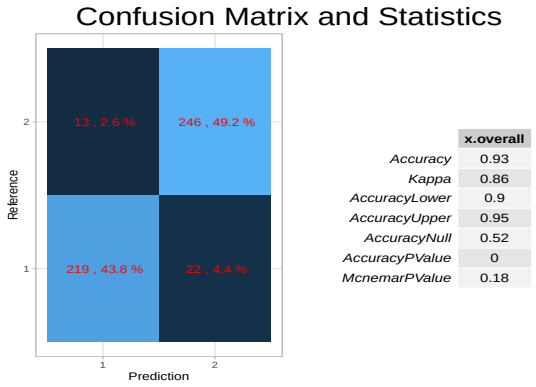


Figure 4.3: The Visualization of Confusion Matrix and Statistics of MCWM with $n = 500$.

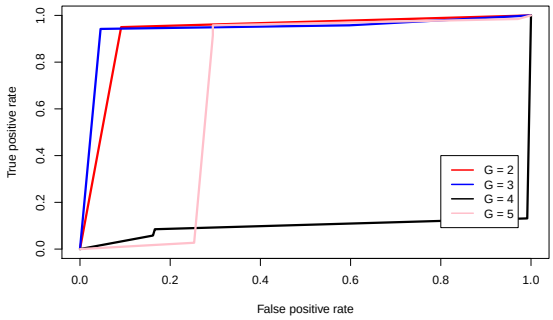


Figure 4.4: The Receiver's Operating Characteristics curve of the prediction by Multinomial Cluster Weighter Model with $n = 500$.

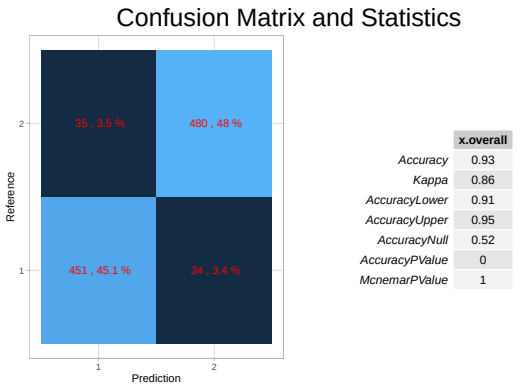


Figure 4.5: The Visualization of Confusion Matrix and Statistics of the Multinomial Cluster Weighter Model with $n = 1000$

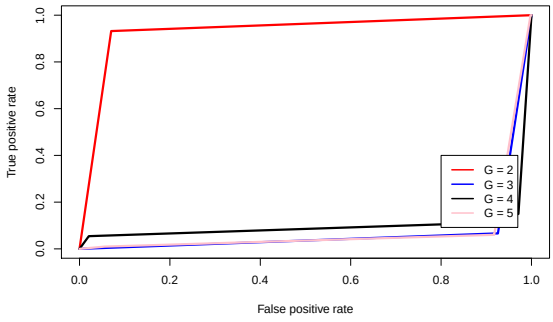


Figure 4.6: The Receiver's Operating Characteristics curve of the prediction by Multinomial Cluster Weighter Model with $n = 1000$.

Figure (4.4) and Figure (4.6) show the Area under Receiver's Operating Characteristic curve of the prediction produced by multinomial CWM for both $n = 500$, and 1000 each with $G = 2, 3, 4, 5$. In (4.4), the two component MCMW $G = 2$ achieves higher accuracy than other groups considered. The area under the curve for $G = 2$ coincides with the area under the curve for the component $G = 3$, this might indicate that many of the classes are distributed across the two groups while only a few observations are clustered in the third group. Four component MCWM $G = 4$ falls below 50% accuracy which indicates an inappropriate model for the data, and the area under ROC curve

Table 4.6: The values of 8 core selected information criteria of MCWM for different $n = 500, 1000$ and $G = 2$. The values in bold face are the smallest values selecting the true values of the artificial data.

n	G	AIC	BIC	ICL	AWE	AIC3	AICc	AICu	Caic
500	**2**	**3641.63**	**3742.78**	**3739.40**	**3223.33**	**3521.63**	**3639.11**	**3613.46**	**3420.48**
	3	3982.60	4134.32	4130.97	3355.15	3802.60	3976.84	3938.40	3650.87
	4	4633.17	4835.47	4831.39	3796.56	4393.17	4622.73	4571.16	4190.86
	5	4744.61	4997.48	4992.71	3698.85	4444.61	4727.93	4662.88	4191.73
1000	**2**	**7031.83**	**7149.62**	**7140.55**	**6580.26**	**6911.83**	**7030.60**	**7005.28**	**6794.05**
	3	9414.11	9590.79	9580.34	8736.75	9234.11	9411.34	9373.64	9057.43
	4	8983.30	9218.87	9209.89	8080.15	8743.30	8978.35	8928.11	8507.73
	5	11155.14	11449.60	11442.99	10026.21	10855.14	11147.34	11084.4	10560.67

Table 4.7: Adjusted Rand Index and its variants of the three-component Model for $n = 500$ and 1000

n	G	Rand	HA	MA	FM	Jaccard	AUC
500	2	**0.870**	**0.739**	**0.739**	**0.870**	**0.769**	**0.930**
	3	0.812	0.624	0.625	0.794	0.648	0.939
	4	0.840	0.680	0.681	0.827	0.697	0.901
	5	0.802	0.604	0.605	0.781	0.627	0.709
1000	2	**0.871**	**0.743**	**0.743**	**0.871**	**0.772**	**0.931**
	3	0.868	0.736	0.736	0.868	0.766	0.929
	4	0.824	0.648	0.649	0.809	0.673	0.900
	5	0.851	0.703	0.703	0.847	0.735	0.925

of five components $G = 5$ is lower than both $G = 2$ and $G = 3$. ROC graphs are two-dimensional graphs in which *tp rate* is plotted on the Y axis and *fp rate* is plotted on the X axis. The point $(0, 0)$ depicts that the classifier never issues a positive classification i.e., it commits no false positive error and also gains no true positives. Conversely, the point $(1, 1)$ is an unconditional issuing of positive classifications. While point $(0, 1)$ is a perfect classifier. In Figure (4.4), $G = 4$ can be thought of as having random performances i.e., they randomly guess the positive class half the time and negative class half the time correctly. Similarly, in Figure (4.6), the components other than two components $G = 2$ perform randomly by guessing true positive and false positive $< 50\%$ of the time. The ROC plot in Figure (4.6) gets improved with large sample size $n = 1000$. There is a clear-cut distinction between

area under the ROC curve of two components and three component when exposed to large datasets.

Table (4.6) shows the values of eight different information criteria described in Table (2.2). We compared different the groups to investigate the identifiability power of the model. It is observed that all the eight model selection criteria agree to the selection of the model with $G = 2$ which aligns to the true component of the model. In Table (4.6), the selection criteria of $G = 2$ for both sample sizes have the smallest values and it shows the model is identifiable for this simulated data. Table (4.7) shows the values for the ARI and its variants to further establish the selection of the true component and performance evaluation of the model. The higher the values, the stronger the agreement between the actual classes and the predicted classes. It can be seen that two-component MCWM has the highest values among other number of components in both sample sizes. This simply means that the model with the components other than the true component performed poorly in this simulation study.

4.5.5 Results for the three components

We present the results for the three-components MCWM $G = 3$ for sample sizes $n = 500$ and $n = 1000$.

In table (4.8), the estimates for the mean vector, the mixing proportion, and the sigma are presented. Also, the estimates of the coefficients are shown in Table (4.9). With three component, MCWM provides good estimates of the parameters for both sample sizes. Table (4.10) shows the result of the component selected by the eight information criteria. It can be seen that all the eight information criteria agree together in both sample sizes. Figure (4.7) and Figure (4.8) presents the plots of the observations, where each observation is clustered by the color of their classes.

In Figure (4.9) and Figure (4.10), we visualize the confusion matrix whose values are presented in Table (4.12). Also, the ROC plots can be visualized in Figure (4.10) and Figure (4.12). According to Figure (4.11) and Figure (4.12) for $n = 500$ and $n = 1000$ respectively, the area under ROC curve of the three-component MCWM is higher than other number of components considered in the study.

Table 4.8: Recovered values $n = 500, 1000$ and $G = 3$

	n	g	$\hat{\mu}_1$	$\hat{\mu}_2$	σ^2_{11}	σ^2_{22}	$\boldsymbol{\pi}$
True		1	0.100	0.000	1.000	1.000	1/3
		2	−2.000	1.000	1.000	1.000	1/3
		3	2.000	3.000	1.000	1.000	1/3
Recovered	500	1	0.031	−0.179	0.880	0.870	0.314
		2	−1.919	1.043	1.110	1.086	0.330
		3	1.954	2.987	1.008	1.011	0.356
Recovered	1000	1	0.235	0.014	0.901	1.049	0.340
		2	−1.976	1.066	0.987	1.040	0.343
		3	2.016	3.110	0.911	0.967	0.317

Table 4.9: Recovered values $n = 500, 1000$ and $G = 3$ with c as the baseline

	n	g	Y	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
True		1	a	0.000	5.000	0.400
			b	0.000	0.300	0.040
		2	a	0.000	0.010	0.020
			b	0.000	2.000	1.000
		3	a	0.000	1.000	0.030
			b	0.000	0.060	0.020
Recovered	500	1	a	0.263	6.236	0.301
			b	0.057	0.714	0.102
		2	a	0.204	0.326	0.093
			b	0.232	2.093	0.651
		3	a	0.051	0.581	0.103
			b	−0.085	0.417	−0.420
Recovered	1000	1	a	0.135	4.710	0.385
			b	−0.152	0.044	−0.060
		2	a	−0.283	−0.087	−0.039
			b	−0.262	1.426	0.805
		3	a	−0.622	1.113	0.132
			b	−0.729	−0.134	0.346

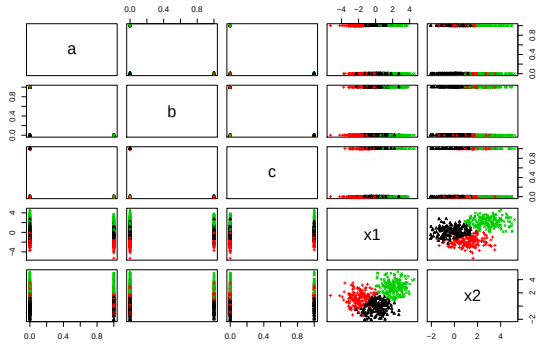


Figure 4.7: The classification plot of MCWM for $n = 500$ with covariates.

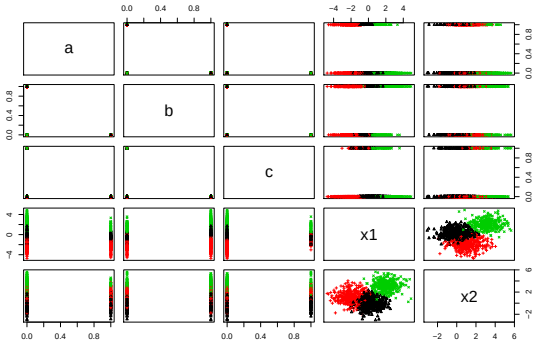
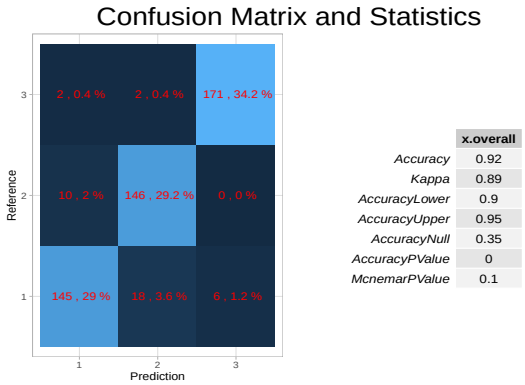


Figure 4.8: The classification plot of MCWM for $n = 1000$ with covariates.



	x.overall
Accuracy	0.92
Kappa	0.89
AccuracyLower	0.9
AccuracyUpper	0.95
AccuracyNull	0.35
AccuracyPValue	0
McnemarPValue	0.1

Figure 4.9: The Visualization of Confusion Matrix and Statistics of the MCWM with $n = 500$.

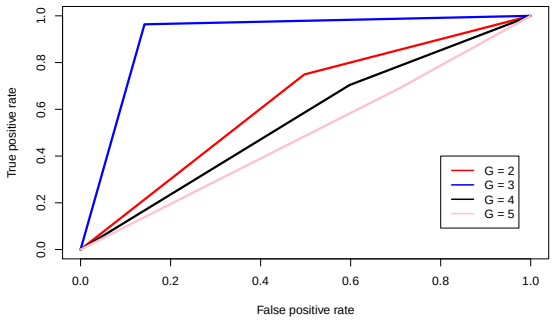
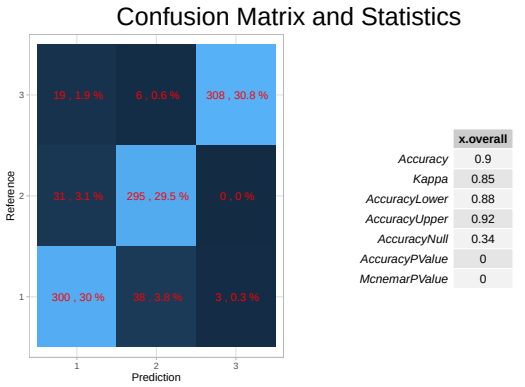


Figure 4.10: The ROC curve of the prediction by MCWM with $n = 500$



	x.overall
Accuracy	0.9
Kappa	0.85
AccuracyLower	0.88
AccuracyUpper	0.92
AccuracyNull	0.34
AccuracyPValue	0
McnemarPValue	0

Figure 4.11: The Visualization of Confusion Matrix and Statistics of the MCWM with $n = 1000$

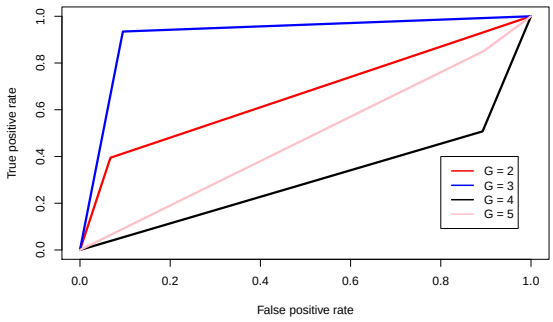


Figure 4.12: The ROC curve of the prediction by MCWM with $n = 1000$.

Table 4.10: The values of Information Criteria of MCWM for different $n = 500, 1000$ and $G = 3$

n	G	AIC	BIC	ICL	AWE	AIC3	AICc	AICu	Caic
500	2	4491.25	4592.40	4589.03	4072.95	4371.25	4488.73	4463.08	4270.10
	3	**3997.76**	**4149.49**	**4143.77**	**3370.31**	**3817.76**	**3992.01**	**3953.57**	**3666.03**
	4	4483.21	4685.51	4681.26	3646.61	4243.21	4471.78	4421.21	4040.91
	5	4703.66	4956.54	4951.74	3657.91	4403.66	4686.99	4621.93	4150.79
1000	2	8459.67	8577.46	8565.33	8008.10	8339.67	8458.44	8433.12	8221.88
	3	**7896.39**	**8073.07**	**8061.30**	**7219.03**	**7716.39**	**7893.62**	**7855.92**	**7539.71**
	4	8528.43	8764.01	8752.98	7625.29	8288.43	8523.49	84.73.25	8052.86
	5	8737.17	9031.64	9022.30	7608.24	8437.17	8729.38	8666.44	8142.71

Table 4.11: Adjustment Rand Index in the three component Model for $n = 500$

n	G	Rand	HA	MA	FM	Jaccard	AUC
500	2	0.543	0.117	0.119	0.499	0.319	0.509
	3	**0.908**	**0.793**	**0.794**	**0.862**	**0.757**	**0.950**
	4	0.804	0.542	0.544	0.685	0.518	0.878
	5	0.802	0.530	0.532	0.672	0.501	0.709
1000	2	0.614	0.273	0.273	0.606	0.410	0.738
	3	**0.882**	**0.734**	**0.734**	**0.822**	**0.696**	**0.930**
	4	0.860	0.675	0.675	0.777	0.633	0.858
	5	0.842	0.613	0.614	0.731	0.559	0.867

Table (4.10) shows the results of the selected eight different information criteria. All the eight information criteria provide a correct selection of the number of components for both 500 and 1000. Table (4.11) shows the values for the ARI and its variants in order to further establish the model performance. Again, the model with $G = 3$ has the highest ARI and AUC values of 0.95 and 0.93 for both sample sizes. In Table (4.12), the confusion matrix is presented for $n = 500$ at the top and $n = 1000$ at the bottom. The overall classification accuracy for $n = 500$ and $n = 1000$ are 92.4% and 90.3% respectively.

Table 4.12: Confusion Matrix in the three component Model for $n = 500$ and 1000

n	Component		1	2	3	MR (%)
500	1		145	18	6	14.20
	2		10	146	0	6.41
	3		2	2	171	2.29
rate		7.60				
Accuracy		92.40%				
1000	1		300	31	10	14.29
	2		38	295	6	12.98
	3		3	0	308	0.96
Misclass		9.70				
Accuracy		90.30%				

4.6 MCMW For Real Moderate Data

4.6.1 The Use of Contraceptive Among married women

The dataset about the use of contraceptive among married women is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or not aware of any pregnancy at the time of interview [Dua & Graff (2017)]. The goal is to predict the choice of the current contraceptive methods (no use, long-term methods, or short-term methods) of a woman based on her demographic and socioeconomic characteristics.

The following information about the couples are given as follows: age: numerical, education: 1 = low, 2, 3, 4 = high, Number of children ever born: (numerical), Wife's religion: 0 = Non-Islam, 1=Islam, Wife's now working: 0=Yes, 1=No, Standard-of-living index: 1=low, 2, 3, 4=high, Media exposure: 0 = Good, 1 = Not good, contraceptive method used (class attribute): 1 = No-use, 2 = Long-term, 3 = Short-term.

First, we performed feature extraction using the Principal component analysis. The principal components value with a cumulative of 0.9 was used to extract the feature. This reduces the features further to two features and it is presented in Figure (4.13). Figure (4.15) shows the cluster plot of the married women using contraceptive.

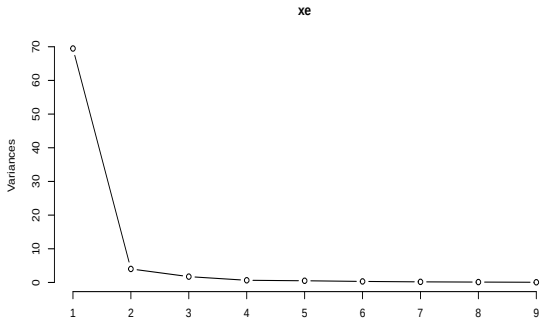


Figure 4.13: The feature extraction selected by principal component analysis for contraceptive. The selection technique is due to the cumulative variance of 90%, i.e. only two features explain about 90% of the variance in the data.

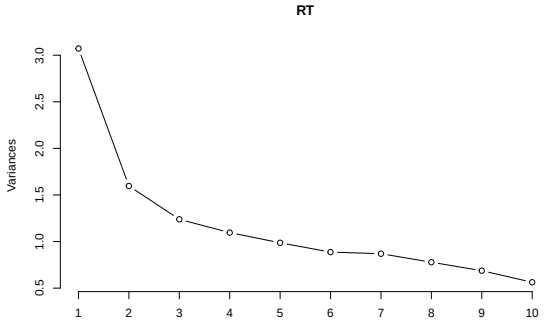


Figure 4.14: The feature extraction selected by principal component analysis for Heart data. The selection technique is due to the cumulative variance of 90%, i.e. only three features explain more than 90% of the variance in the data.

Table 4.13: Confusion Matrix of the three-component MCWM for the use of contraceptives among married woman. MCWM has the highest prediction accuracy of about 99%.

Real Component	1	2	3	Misclassification rate (%)
1	625	4	0	0.64
2	0	316	17	5.11
3	0	0	511	0.00
Misclassification	1.43			
Accuracy	98.57%			

In group one the number of married women according to the prediction of the model is 625, while the actual number in cluster one is 629. There is mild misclassification in cluster one. The cluster two according to MCWM has 316 married women. These are correctly classified as the married women that have long-term use of contraceptive while the prediction of married women in cluster three is 511. Multinomial CWM correctly predicts that 511 married women have a short-term use of contraceptive.

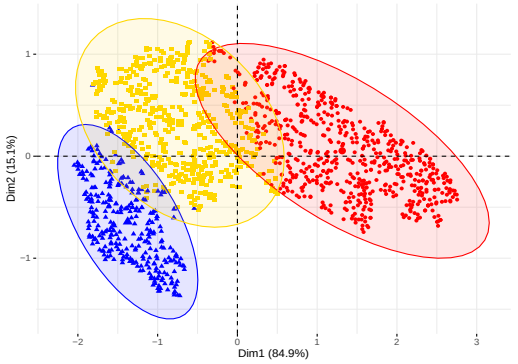


Figure 4.15: The cluster plot of the married women using contraceptive with three levels. Cluster 1 has 625 married women, cluster 2 has 316 married women, and cluster 3 has 511 married women.

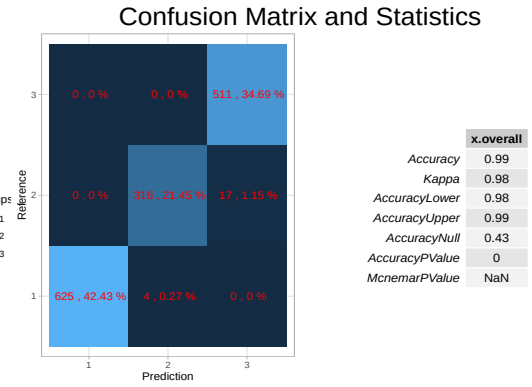


Figure 4.16: The Visualization of Confusion Matrix and Statistics of the MCWM prediction of the use of contraceptives among married women. The result shows low confusion in the prediction of the MCWM model.

4.6.2 Heart Data from Cleveland database

The Cleveland database has been used widely by many researchers such as Detrano et al. (1989); David & Dennis (1988); Gennari et al. (1989). The goal refers to the presence of heart disease in the patient. This database contains 76 attributes from which many experiments have suggested using subset of 14 attributes. The class label is categorical where 0 represents absence of heart disease and $1, 2, 3, 4$ represents the "presence" of heart disease in the patient. Although there are five classes but the intrinsic cluster is binary. Our goal is to investigate the discovering power of the proposed model in discovering the hidden cluster among the patients.

Again we first carried out a feature extraction mechanics by transforming using the PCA. Figure (4.14) shows that three features account for more than 90% of the spread in the data. The essence of PCA is also to de-correlate the features. Afterwards, we assumed that there are five classes and performed a model selection based on the number of mixing components in the data since the response variable Y has five levels, anyone might be tempted to assume that there are four or at most five groups. However, Table (4.14) shows that all the eight information criteria revealed the hidden cluster of $G = 2$ rather than $G = 5$. This supports the claim of other experiments with the Cleveland database which have attempted to distinguish presence

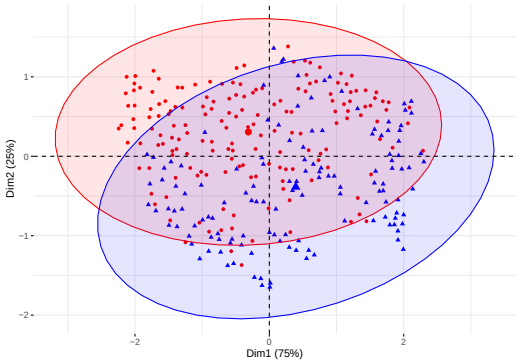


Figure 4.17: The cluster plot of the patients diagnosed with heart disease: The class is absence (0) and presence (1,2,3,4).

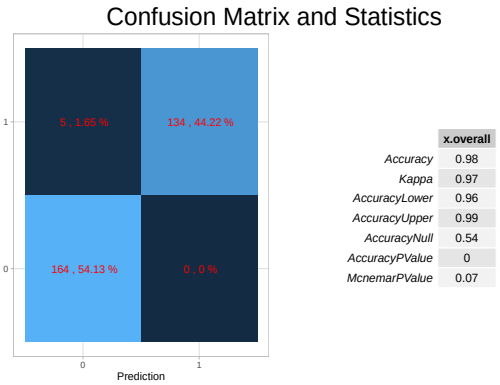


Figure 4.18: The Visualization of Confusion Matrix and Statistics of the MCWM prediction of the diagnosis of the patient with heart disease.

(values: $1, 2, 3, 4$) from absence (value: 0). From Figure (4.18), it can be shown that multinomial CWM misclassified 5 patients wrongly and achieves about 98% overall. We note here that there is a difference between supervised learning and supervised clustering. Supervised learning focuses in the class label and the predicted class, while the supervised clustering focuses on the location of the observations hereby called clusters. To perform confusion matrix, we "binarize" the class label with presence = 1 and absence = 0. This will enhance a comparison between the actual class and the predicted. We also sorted both classes for proper comparison. Table (4.14) shows the choice of number of component in the heart data. All the eight information criteria select the number of component $G = 2$. This confirms what other researchers have done to increase the prediction accuracy of their model on the data.

Table 4.14: The values of eight choices of Information Criteria of MCWM for different mixture component G. The outcome has five levels which is absence (0) and presence $(1, 2, 3, 4)$. According to the information criteria, the number of mixture component selected is $G = 2$ which confirms what previous researchers naively suggested.

G	AIC	BIC	ICL	AWE	AIC3	AICc	AICu	Caic
2	**1358.71**	**1440.41**	**1435.48**	**997.30**	**1248.71**	**1355.09**	**1331.17**	**1167.01**
3	1652.25	1774.81	1773.41	1110.15	1487.25	1643.91	1607.85	1364.70
4	2085.84	2249.24	2245.28	1363.03	1865.84	2070.49	2021.78	1702.43
5	1982.89	2187.15	2186.65	1079.38	1707.89	1957.95	1896.04	1503.64

Table 4.15: Confusion Matrix in the three component Model for the multinomial CWM compared with other models used for the heart disease data. Among these model, MCWM has the highest prediction accuracy of about 99%

	Component	1	2	MR (%)
	1	164	5	7.20
	2	0	134	6.61
Multinomial CWM	Logistic-Regression	NTgrowth	C4	CLASSIT
98.35%	77%	77%	74.8%	78.9%

Table (4.15) shows the prediction accuracy of the proposed model MCWM. It can be seen that previous models such as Logistic regression and NTgrowth model achieved 77%, C4 has a prediction accuracy of 74.8%, and the CLASSIT model has 78.9% accuracy. Figure (4.17) show the cluster of the patient diagnosed with heart disease and without heart disease. Cluster one has 164 patients with no presence of heart disease while cluster two has 134 patients with the presence of heart disease. The actual data have 169 patients with no heart disease and 134 patients with heart disease.

4.7 MCWMs for Real High-dimensional data

In this section, we compare the MCWM with model-based clustering (Mclust) [Scrucca et al. (2016)] and High Dimensional Data Clustering (HDDC) [Bouveyron et al. (2007)].

The MNIST dataset comprising of 10-class handwritten digits, was first introduced by LeCun et al. (1998). The full handwritten digit data is divided into training set and test set. The size of training set is $60,000 \times 784$ and the test set is $10,000 \times 784$. First, we use USPS358 data set. However, due to large volume of full handwriting data set, Mclust and HDDC are not used for the full handwritten image data. This shows the limitations of the EM algorithm which is the inability for scalability to high-dimensional data.

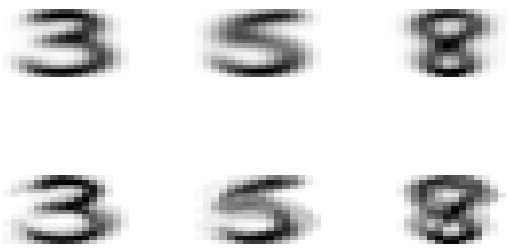


Figure 4.19: The original image of the data of the digits 3, 5 and 8 to be recognized is presented at the top while the means of the posterior produced by MCWM is presented at the bottom.

4.7.1 USPS358 Data set

The USPS358 data contain only the $1,756$ images of the digits $3, 5$ and 8 which are the most difficult digits to discriminate. Each digit is a 16×16 gray level image and is represented as a 256-dimensional vector in the USPS358 data set.

Figure (4.19 top) shows a sample of handwritten digits from the USPS postal services (USPS358 data set in the **MBCbookR** package). And Figure (4.19 below) are the images of the handwritten number of the different handwriting from the means of the result produced by MCWM. We note here that we have used cluster and class interchangeably. Table (4.16) shows the confusion matrix, Accuracy, and Adjusted Rand Index. It can be seen that the proposed model MCMC outperforms other competing models such as Mclust and HDDC. The Accuracy of MCWM is 100% while Mclust achieves 31.89% and HDDC achieves 35.14%. This is also confirmed by the result of the Adjusted Rand Index that counts the number of correctly classification between the actual and the predicted.

Table 4.16: Confusion Matrix, Accuracy, and Adjusted Rand Index of the MCWM compared with other models used for the USPS data. Among these model, MCWM has the highest prediction accuracy of about 100%.

Real Component	3	5	8
3	658	0	0
5	0	556	0
8	0	0	542
Accuracy	MCWM	Mclust	HDDC
	100.00%	31.89%	35.14%
ARI	MCWM	Mclust	HDDC
	100.00%	63.42%	80.50%

4.7.2 Full Handwritten digit image

We performed an analysis on the full image of the hanwritten digits. Table (4.17) shows the results of the confusion matrix for both training set and test set. In Table (4.17), the accuracy provided by MCWM is about 93%, while MCWM achieved the prediction accuracy of about 92%. The minimized difference between the training result and the test result shows that there is no presence of overfitting. We note here that no comparison was made with both Mclust and HDDC due to the volume of the data. The use of batch size technique makes MCWM scalable to high-dimensional data. Also, we computed the Adjusted Rand Index for the training set and test set to be 81% and 80% respectively.

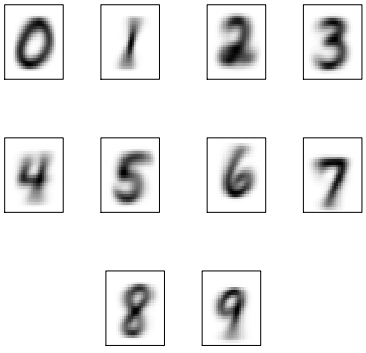


Figure 4.20: Image recognized from 10 groups

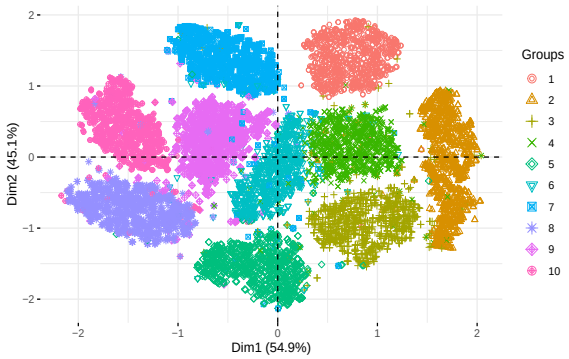


Figure 4.21: The MCWM classification plot for MNIST data set

Figure (4.20) shows a well partitioned classes of the numbers which is the output of MCWM. Finally, Figure (4.21) shows the plot of each class in low-dimensional space.

Table 4.17: Confusion Matrix of the MCWM used for the training set and test set of Handwritten Image

		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
Training	c1	5807	1	14	20	8	19	8	8	28	10
	c2	1	6629	19	22	2	13	1	17	23	15
	c3	25	82	5298	178	61	28	45	74	137	30
	c4	19	19	67	5673	6	178	5	46	61	57
	c5	26	26	26	10	5383	10	26	31	25	279
	c6	53	25	34	236	29	4820	49	18	120	37
	c7	73	26	69	16	52	99	5549	8	19	7
	c8	17	16	42	26	31	10	1	5965	6	151
	c9	24	200	45	328	16	273	28	29	4784	124
	c10	25	20	17	54	87	42	2	194	16	5492
		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
Testing	c1	964	0	1	2	1	4	3	3	1	1
	c2	0	1120	2	4	0	1	3	2	3	0
	c3	8	19	898	35	10	3	11	14	31	3
	c4	4	0	6	942	1	28	1	12	8	8
	c5	1	2	6	4	897	0	5	10	10	47
	c6	11	4	3	51	6	769	9	7	25	7
	c7	17	3	10	6	11	30	880	1	0	0
	c8	2	7	16	9	3	1	0	957	2	31
	c9	9	22	5	46	6	64	8	16	780	18
	c10	10	8	0	10	15	10	0	31	1	924

4.8 Summary

In this Chapter, we have introduced a new cluster weighted models called a Multinomial CWM for clustering data with multiclass response variables by introducing the softmax function as the probability of the multinomial distribution. Multinomial CWM extends preexisting Binomial CWM for binary response variables [Ingrassia et al. (2015)]. Different from the previous work in the field of cluster weighted model, MCWM allows modeling of multinomial response variables. Furthermore, we de-

scribed through simulation study an EM algorithm via both Iteratively Re-weighted Least Squares and Stochastic Gradient Descent for parameter estimation. To investigate the performance of the model, several performance metrics was used such as confusion matrix, Receiver's Operating Characteristics (ROC), and ARI with its variants. We also showed through different eight information criteria that the MCWM is able to discover the mixture component hidden in the data. Following the conditions of identifiability, we can say that MCWM is identifiable. We provided different supporting plots such as the classification plots, confusion matrix plots, and ROC plot. We also evaluated the performance of the proposed model on two real datasets. It was shown that Multinomial CWM has higher accuracy when compared to other models used for the data.

However, from the perspective of the EM-IRLS algorithm for the parameter estimation of MCWM, the main limitations encountered are general limitations arising from EM algorithm such as in-scalability to large dataset due to slow-to-convergence nature of EM algorithm and the problem of initialization. The matrix inversion from the multinomial distribution estimation computed by the iteratively re-weighted least squares could lead to the problem of singularity in the covariance matrix. This problem of singularity at the M-step often causes the EM algorithm to degenerate. Additionally, the use of conventional maximum likelihood techniques at the M-step such as **optim** function and **multinom** in R [R (2019)] become so unstable due to covariance matrix inversion. We used more stable version of ML technique in **R** [R (2019)] which is the **optim_sa** function from the **R** package **Optimization**. In the future, we wish to explore different variants of EM algorithm for this type of proposed model such as EM algorithm via Simulated Annealing (SA). EM-SA algorithm will avoid the problem of local maxima in EM algorithm. With the help of EM-SGD which we have established, we also wish in the future to work in the perspective of Deep Learning where MCWM will be computed at each hidden unit. Moreover, another possible future direction is to investigate different cross-validation techniques and regularization methods.

In the next chapter, we will address the problem of large proportion of zeros in the class label by proposing a CWMs that can account for the presence of large proportion of zeros in the class label which often causes misleading or erroneous

interpretation or inference for decision making. This problem is generally known as class imbalance in the data. For example, defects in manufacturing usually occur when the manufacturing equipment is not properly aligned. If the equipment is misaligned, defect can occur according to Poisson distribution. This means that the defects in manufacturing occurs with inflation at zero. Similarly, in medical data, the record of a particular patient can be censored or missing before the end of a follow-up study.

Chapter 5

Variational Bayesian: EM – IRLS Zero-Inflated Poisson CWM

5.1 Introduction

Two-component mixture models are frequently used to model data that contain excess zeros. In medical context, a possibility of excess zeros might be due to the fact that the patient is cured after the treatment and no realization of the symptom being monitored in a follow-up will occur. This phenomenon can be handled by a two-component mixture where one of the components is taken to be a degenerate distribution, having mass 1 at $y_j = 0$. While the other component is a Poisson (or binomial) regression model depending on the situation.

A Finite mixture of Poisson regression models with constant weights parameters have been developed by Wedel et al. (1993); Brannas & Rosenqvist (1994); Wang et al. (1996); and Alfo & Trovato (2004). Wang et al. (1998) incorporated covariates in the weight parameters of finite mixture Poisson regression models, treating the covariates as the concomitant variables. To account for different covariates with count response variables, Ingrassia et al. (2015) proposed a generalized linear mixed cluster-weighted model where the response variable is allowed to follow the Poison distribution. As an alternative to Poisson regression model for handling over-dispersion in data, a Negative binomial (NB) regression model can be used. The count variable of interest may contain more zeros than expected under a Poisson model, which is observed in

many applications. Inflated zeros can cause instability in a predictive model. Class imbalance also can be a consequence of zero inflation in a response variable we wish to predict therefore causing irregularities in the prediction accuracy and result in overfitting. Zero-Inflated Poisson (ZIP) regression model has been proposed with an application to defects in manufacturing [Lambert (1992)]. The ZIP distribution is a mixture of a Poisson and a degenerate distribution at zero. This regression setting allows for the covariates in both Poisson mean and weight parameter. Furthermore, in a situation where over-dispersion takes precedence, a zero-inflated negative binomial (ZINB) regression model can be a better fit. However, if a population has excess zeros and several sub-populations in non-zero counts, a single component of the ZINB regression model may fail to capture the excess or sufficient to describe the non-zero counts.

5.1.1 Main Contribution

In this Chapter, we wish to address the problem of class imbalance by proposing a zero-inflated Poisson cluster weighted (ZIPCWM) model that is capable of handling zero-inflation in data. ZIPCWM extends Poisson cluster weighted models and other mixture models. Moreover, ZIPCWM model allows for a mixed covariates to be either discrete or continuous or both. Contrary to the existing zero-inflated models, ZIPCWM can be used as a classification model which is appropriate in handling uneven class distributions. We investigate further the effect of mixed covariates in the zero-inflated Poisson cluster weighted models. To estimate the parameter of the models, we propose an Expectation-Maximization (EM) algorithm via an iteratively reweighted least squares for ZIPCWM. We analytically investigate the identifiability of the proposed model through an extensive simulation study. Parameter recovery, classification assessment, and performance of different information criteria are investigated through broad simulation design. The ZIPCWM is applied to real data which accounts for excess zeros of over 40%. We explore the classification performance of ZIPCWM, Fixed Zero-inflated Poisson mixture model (FZIP), and Poisson cluster weighted model (PCWM) on a real data. Furthermore, we will compare the classification strength of the new proposed with its existing mixture models such as Poisson cluster weighted model and Fixed zero-inflated Poisson mixture models.

5.1.2 Mixed Continuous and Categorical Variables

We consider now the problem of fitting a family of cluster weighted model

$$p(\boldsymbol{x}, y) = \sum_{g=1}^{G} = \pi_g p(y|\boldsymbol{x}, \mathcal{D}_g) p(\boldsymbol{x}|\mathcal{D}_g) \tag{5.1}$$

to some pair of data $(\mathbf{X}', Y)'$, where $\mathbf{X}$ is a matrix of covariates, and Y is a random variable defined on some space $\mathcal{D}$ and where some of the variables are categorical. Basically, we can assume that the categorical variables are independent of each other and the continuous variables which can be taken to have any continuous distribution such as the multivariate normal distribution. This idea of mixed covariates has come into prominence more recently due to its appearance in the graphical modeling of mixed variables known as the conditional Gaussian distribution model [Whittaker (1990); Cox & Wermuth (1992)]. The location model has been used for fitting the mixture models to mixed categorical and continuous variables, [Jorgensen & Hunt (1996); Lawrence & Krzanowski (1996); and Lawrence & Krzanowski (1999)].

5.2 The Zero-Inflated Poisson CWM.

Suppose that $\mathbf{X}$ can be decomposed as $(\mathbf{Q}', \mathbf{W})'$, where $\mathbf{Q}$ is a q-variate vector of continuous covariates and $\mathbf{W}$ is a p-variate vector of categorical variable respectively, being $d = q + p$. In this case, $\mathcal{X} = \mathbf{R}^{p \times \{1,\ldots,r_1\},\times,\ldots,\times\{1,\ldots,r_p\}}$. With the location model, the $\mathbf{W}$ categorical variables are replaced by a single multinomial random variable $\mathbf{W}_i$ with p cells.

Suppose that Y is a count response variable, then the probability of observing zeros is π_1, and the probability of observing Poisson cluster weighted model (PCWMs) is $1 - \pi_1$. The ZIP cluster weighted model is given as follows:

$$p(\boldsymbol{x}, y; \boldsymbol{\Theta}) = \pi_1 I_{(y=0)} + \sum_{g=2}^{G} p\Big(y\Big|\boldsymbol{x}; \boldsymbol{\zeta}_g\Big) p\Big(\boldsymbol{x}; \boldsymbol{\psi}_g\Big) \pi_g, \tag{5.2}$$

where Equation (5.2) is decomposed into categorical and continuous covariates as follows;

$$= \pi_1 I_{(y=0)} + \sum_{g=2}^{G} p\Big(y\Big|\boldsymbol{x};\boldsymbol{\zeta}_g\Big)p\Big(\boldsymbol{q};\boldsymbol{\psi}_g^*\Big)p\Big(\boldsymbol{w};\boldsymbol{\psi}_g^{**}\Big)\pi_g \tag{5.3}$$

since $y = 0$, and its mean $\mu_{i1}(\boldsymbol{x};\boldsymbol{\beta}_1) = 0$, then $p(y|\boldsymbol{x};\boldsymbol{\zeta}_1) = 1$. If we assume that the response variable contains many zeroes, then $p(y|\boldsymbol{x};\zeta_g)$ is modeled as a Zero-Inflated Poisson regression, $p(\boldsymbol{w}|\boldsymbol{\psi}_g^{**})$ follows a multinomial distribution and $p(\boldsymbol{q}|\boldsymbol{\psi}_g^{*})$ follows a Gaussian distribution, where G is the number of mixing components, π_g is the mixing weight of component g such that $0 < \pi_g < 1$, $g = 1, ..., G$ and $\sum_{g=1}^{G}\pi_g = 1$ by the decomposition $\pi_1 = 1 - \sum_{g=2}^{G}\pi_g$. The weight π_1 determines the proportion of excess zeros compared with an ordinary Poisson mixture model determined by $1 - \pi_1$.

5.2.1 Modeling for $\mathbf{p}(y|\boldsymbol{X};\boldsymbol{\eta}_g)$

To handle response variable with excess zeros, we assume that the density of $p(y|\boldsymbol{x};\boldsymbol{\eta}_g)$ belong to exponential family. It is a common practice that the exponential family is strictly related to the generalized linear models with a monotone and differential link function $f(.)$ that makes the expected value μ_g, of $Y|\mathcal{D}_g$ depend on the covariate $\mathbf{X}$ through the linear combination $f(\mu_g) = \boldsymbol{x}'\boldsymbol{\beta}_g$. Note here that we have not used $\boldsymbol{\beta}_g$ explicitly, so it contains the coefficient and $\mathbf{X}$ will have extra column of ones. The distribution of $Y|\mathcal{D}_g$ is denoted by $p(y|\boldsymbol{x};\boldsymbol{\beta}_g)$. Assuming that Y takes values in $\mathcal{Y}$ and that the conditional density $Y|\boldsymbol{x},\mathcal{D}_g$ is Poisson with parameter $\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)$; that is, $Y|\boldsymbol{x},\mathcal{D}_g \sim P[\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)]$. We allow the mean of the response variable to depend on the covariates $\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)$ using the following regression models that is,

$$\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g) = \exp(\boldsymbol{x}_i'\boldsymbol{\beta}_g), \quad i = 1, ..., N, g = 2, ..., G \tag{5.4}$$

In this case,

$$p(y|\boldsymbol{x};\boldsymbol{\beta}_g) = \exp[-\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)]\frac{[-\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)]^y}{y!} \tag{5.5}$$

The mixing weights $\{\pi_g\}_{g=1}^{G}$ can be treated as the multinomial logit of π_g to be a linear function of covariates which is commonly known in the literature as a concomitant

variable.

$$\pi_{ig}(\boldsymbol{v}_i, \gamma) = \frac{\exp(\boldsymbol{v}_i\gamma_g)}{\sum_{g=1}^{G}\exp(\boldsymbol{v}_i\gamma_g)} \tag{5.6}$$

In this case, $\mathbf{x}_i = (x_{i1}, ..., x_{ip})$ and $\boldsymbol{v}_i = (v_{i1}, ..., v_{ir})$ are $1 \times p$ and $1 \times q$ rows vectors of covariates (including an intercept), respectively. They can be the same or have nothing in common. The regression coefficient for the gth component are β_g and γ_g which are vectors of $p \times 1$ and $q \times 1$ respectively. We note that the mixing probability of the first component $\pi_{i1}(\boldsymbol{v}_i, \gamma)$ is taken to be the baseline for the multinomial logistic model.

5.2.2 Modeling $p(\boldsymbol{x}; \boldsymbol{\psi}_g)$

The term $p(\boldsymbol{q}; \boldsymbol{\psi}_g^*)$ in Equation (5.3) is modeled as a q-variate Gaussian density with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, i.e., $p(\boldsymbol{q}|\boldsymbol{\psi}_g^*) = \phi(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. With respect to the term $p(\boldsymbol{w}; \boldsymbol{\psi}_g^{**})$ in Equation 5.3, we assume that each categorical covariate can be taken to be a binary vector $\boldsymbol{w}^k = (w^{k1}, ..., w^{kr_k})'$, where $w^{ks} = 1$ if r_k is equal to the value s, with $s \in \{1, ..., r_k\}$, and $w^{ks} = 0$ otherwise. Furthermore, we assume that q categorical covariates are independent of each other. Then,

$$p(w; \boldsymbol{\alpha}_g) = \prod_{k=1}^{p}\prod_{s=1}^{r_k}(\alpha_{gks})^{w^{ks}}, \tag{5.7}$$

where $g = 1, ..., G$, $\boldsymbol{\alpha}_g = (\boldsymbol{\alpha}'_{g1}, ..., \boldsymbol{\alpha}'_{gp})'$ and $\boldsymbol{\alpha}_{gk} = (\alpha_{gk1}, ..., \alpha_{gkr_p})'$. We take the density of $p(\boldsymbol{w}; \boldsymbol{\alpha}_g)$ in Equation 5.7 to be a multinomial distribution of parameters $\boldsymbol{\alpha}_{gk}$, where $k = 1, ..., p$ and $\sum_{s=1}^{r_k}\alpha_{gks} = 1$ with constraint $\alpha_{gks} > 0$.

5.2.3 The Resulting Overall Model

The ZIPCWM over all observation can be formulated as follows:

$$p(\boldsymbol{x}, y; \boldsymbol{\Theta}) = \pi_1 I_{(y=0)} + \sum_{g=2}^{G}\text{Pois}\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\phi\Big(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big)p\Big(\boldsymbol{w}; \boldsymbol{\alpha}_g\Big)\pi_g \tag{5.8}$$

where $I_{(.)}$ is an indicator function that outputs 1 when the specified condition is satisfied and 0 otherwise, and Pois(.) denotes the probability mass function of y_i and

x_i with a mean of $\mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)$. Thus the posterior probability of Equation 5.3 is

$$p(\mathcal{D}_g|\mathbf{y}_i, \mathbf{x}_i) = \frac{\text{Pois}\Big(y_i\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\left(\binom{M_i}{w_{i1},...,w_{is}}\alpha_{ksg}^{w_{i1}}...\alpha_{ksg}^{w_{is}}\right)\mathcal{N}\Big(\mathbf{q}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}\Big)\pi_g}{p(\boldsymbol{x}, y; \boldsymbol{\Theta})} \tag{5.9}$$

5.2.4 Identifiability

To be reliably estimate the parameters of Equation (5.8), we require that the ZIPCWM be identifiable, that is, two sets of parameters which do not agree after permutation cannot produce the same mixture distribution. Teicher (1961) proves that the class of finite mixtures of Poisson distribution is identifiable without covariates. Similarly, Follmann & Lambert (1991) give sufficient conditions for identifiability, and Wang et al. (1996) extend the definition of identifiability of finite Poisson mixtures with covariates. We provide the condition for identifiability of ZIPCWM as follows.

Definition

Consider the collection of probability models $\{p(\boldsymbol{x}_1, y_1, \boldsymbol{\Theta}), ..., p(\boldsymbol{x}_N, y_N; \boldsymbol{\Theta})\}$, with a restriction that $\pi_1 < ... < \pi_g$, sample space $\mathcal{Y}_1, ..., \mathcal{Y}_N$, parameter space $\boldsymbol{\Theta}$, and fixed covariates vectors $x_1, ..., x_N$ that is decomposed to categorical w and continuous variables q, where $w_i \in \mathcal{R}^p$ and $q_i \in \mathcal{R}^d$ for $i = 1, ..., N$. The collection of probability model is *identifiable* if for $(\boldsymbol{\Omega}, \boldsymbol{\pi}, \boldsymbol{k}, \boldsymbol{h})$, $(\boldsymbol{\Omega}^*, \boldsymbol{\pi}^*, \boldsymbol{k}^*, \boldsymbol{h}^*) \in \boldsymbol{\Theta}$,

$$p(\boldsymbol{x}, y, \boldsymbol{\Omega}, \boldsymbol{\pi}, \boldsymbol{k}, \boldsymbol{h}) = p(\boldsymbol{x}, y, \boldsymbol{\Omega}^*, \boldsymbol{\pi}^*, \boldsymbol{k}^*, \boldsymbol{h}^*) \tag{5.10}$$

for all $y_i \in \mathcal{Y}_i, i = 1, ..., N$, implies that $(\boldsymbol{\Omega}, \boldsymbol{\pi}, \boldsymbol{k}, \boldsymbol{h}) = (\boldsymbol{\Omega}^*, \boldsymbol{\pi}^*, \boldsymbol{k}^*, \boldsymbol{h}^*)$.
We note here that the restriction on the mixing probability is a sufficient condition for label switching problems and it means that two models are equivalent if they agree up to permutation of parameters.
We now provide a sufficient condition for identifiability. Suppose that $(\boldsymbol{\Omega}, \boldsymbol{\pi}, \boldsymbol{k}, \boldsymbol{h})$, $(\boldsymbol{\Omega}^*, \boldsymbol{\pi}^*, \boldsymbol{k}^*, \boldsymbol{h}^*)$ satisfy Equation (5.10).

It then implies that

$$\pi_1 I_{(y=0)} + \sum_{g=2}^{G} \text{Pois}\Big(y\Big|\boldsymbol{x};\boldsymbol{\beta}_g\Big)\phi\Big(\boldsymbol{q};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)p\Big(\boldsymbol{w};\boldsymbol{\alpha}_g\Big)\pi_g =$$

$$\pi_1^* I_{(y=0)} + \sum_{c=2}^{C} \text{Pois}\Big(y\Big|\boldsymbol{x};\boldsymbol{\beta}_c^*\Big)\phi\Big(\boldsymbol{q};\boldsymbol{\mu}_c^*,\boldsymbol{\Sigma}_c^*\Big)p\Big(\boldsymbol{w};\boldsymbol{\alpha}_c^*\Big)\pi_c^* \tag{5.11}$$

for each i and $y_i \in \mathcal{Y}_i, i = 1, ..., N$. Teicher's and Hennig's results imply that

$$G = C,\ \pi_g = \pi_c^*,\ \boldsymbol{\beta}_g = \boldsymbol{\beta}_c^*,\ \boldsymbol{\mu}_g^* = \boldsymbol{\mu}_c^*,$$

$$\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}_c^*, \text{ and } \boldsymbol{\alpha}_g = \boldsymbol{\alpha}_c^*$$

for $i = 1, ..., N$ $g = 1, ..., G$, and $c = 1, ..., C$. By definition of ZIPCWM, we obtain

$$\exp(\boldsymbol{x}_i'\boldsymbol{\beta}_g) = \exp(\boldsymbol{x}_i'\boldsymbol{\beta}_c^*), \tag{5.12}$$

Equation 5.12 means

$$(\boldsymbol{\beta}_g - \boldsymbol{\beta}_c^*)'\boldsymbol{x}_i = 0, \text{ for } i = 1, ..., N. \tag{5.13}$$

Then, we say

$$\text{Pois}\Big(y\Big|\boldsymbol{x};\boldsymbol{\beta}_g\Big) = \text{Pois}\Big(y\Big|\boldsymbol{x};\boldsymbol{\beta}_c^*\Big), \phi\Big(\boldsymbol{q};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big) = \phi\Big(\boldsymbol{q};\boldsymbol{\mu}_c^*,\boldsymbol{\Sigma}_c^*\Big),$$

$$p\Big(\boldsymbol{w};\boldsymbol{\alpha}_g\Big) = p\Big(\boldsymbol{w};\boldsymbol{\alpha}_c^*\Big), \pi_g = \pi_c^*. \tag{5.14}$$

Hence a sufficient condition for identifiability is that $\boldsymbol{X}$ is a full rank matrix, where $\boldsymbol{X} = \boldsymbol{x}_1, ..., \boldsymbol{x}_N$.

5.3 Related Mixture Model

This section highlights the special cases of the proposed model. We show a list of special cases of ZIPCWM below;

5.3.1 Poisson CWM

Let $p(\boldsymbol{x}, y; \boldsymbol{\Theta})$ in Equation (5.8) be a ZIPCWM. Assuming that Y takes values y $= \mathcal{N}$, $Y|\boldsymbol{x}; \mathcal{D}_g$ is Poisson regression model with parameter $\mu_g(\boldsymbol{x}; \boldsymbol{\beta}_g)$ and the $\mathbf{X}|\mathcal{D}_g$ with parameter $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. If there is no account of excess zeros, then $Y|\boldsymbol{x}, \mathcal{D}_g$ is modeled with Poisson regression and the $X|\mathcal{D}_g$ is taken to be the Gaussian distribution, then ZIPCWM reduces to Poisson CWM as follows;

$$p(\boldsymbol{x}, y; \boldsymbol{\Theta}) = \sum_{g=1}^{G} Pois\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\phi\Big(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big)p\Big(\boldsymbol{w}; \boldsymbol{\alpha}_g\Big)\pi_g. \tag{5.15}$$

Poisson CWM is proposed by Ingrassia et al. (2015)

5.3.2 Generalized ZIP Regression mixture model

If $p(\mathbf{x}) = 1$ in Equation (5.8), then ZIPCWM reduces to Generalized ZIP mixture distribution if the mixing weight depends on a concomitant variables

$$p(y; \boldsymbol{\Theta}) = \pi_{i1}(\boldsymbol{v}_i, \gamma)I_{(y=0)} + \sum_{g=2}^{G} Pois\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\pi_{ig}(\boldsymbol{v}_i, \gamma) \tag{5.16}$$

and Fixed ZIP if the mixing weight is fixed.

$$p(y; \boldsymbol{\Theta}) = \pi_1 I_{(y=0)} + \sum_{g=2}^{G} Pois\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\pi_g \tag{5.17}$$

Both GZIP and FZIP are proposed by Hwa et al. (2014)

5.3.3 Zero-Inflated Poisson distribution

If $p(\mathbf{x}) = 1$ and $G = 2$ in Equation (5.8), then ZIPCWM reduces to ZIP distribution [Lambert (1992)].

5.3.4 Standard Poisson mixture model:

If both π_g and $\mu_g(\mathbf{x}_i, \beta_g)$ are constant functions and $p(\mathbf{x}) = 1$, ZIPCWM reduces to the standard Poisson mixture model, denoted by

$$p(Y = y_i) = \sum_{g=1}^{G} \pi_g Pois(y_i|\lambda_g) \tag{5.18}$$

where the constancy of $\mu_{ig}(\mathbf{x}_i, \beta_g)$ is taken to be λ_g. It should be noted that we have used λ_g to preserve the original parameter notation of the standard Poisson mixture model. In the following, we present the derivative of an estimation method based on the EM algorithm for the ZIPCWM model described in Equation (5.9).

5.4 Model Estimation by EM-IRLS Algorithm

Let $(\boldsymbol{x}_i', y_i), ..., (\boldsymbol{x}_n', y_n)'$ be a sample of n independent observation pairs drawn from model Equation (5.9). The corresponding likelihood, for a fixed number of component G is given by

$$L(\boldsymbol{\Theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i, y_i; \boldsymbol{\Theta}) = \prod_{i=1}^{N} \Bigg[\pi_1 I_{(y=0)} + \sum_{g=2}^{G} \text{Pois}\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big) \phi\Big(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big) p\Big(\boldsymbol{w}; \boldsymbol{\alpha}_g\Big) \pi_g \Bigg] \tag{5.19}$$

Here, we assume that the number of G is fixed and known a priori and $\mathbf{z}_i = (z_{i1}, ..., z_{iG})'$ be the latent vector of component indicator variables, where $z_{ig} = 1$ if ith subject $(\boldsymbol{x}_i', y_i)'$ comes from $\mathcal{D}_g$ belongs to the gth latent group and $z_{ig} = 0$ otherwise. By assumption, each belongs to one of the G unobservable components therefore, considered missing or incomplete. Using a multinomial distribution for the unobserved vector $\mathbf{z}_i$,where $\{(\boldsymbol{x}_i', y_i, \boldsymbol{z}_i')'; i = 1, ..., N\}$ is the complete data. The the complete-data log-likelihood can be written as

$$L_{\boldsymbol{c}}(\boldsymbol{\Theta}) = \prod_{i=1}^{N} \Bigg[\pi_1 I_{(y=0)} + \prod_{g=2}^{G} \text{Pois}\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big) \phi\Big(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big) p\Big(\boldsymbol{w}; \boldsymbol{\alpha}_g\Big) \pi_g \Bigg]^{z_{ig}} \tag{5.20}$$

The corresponding complete-data log-likelihood by taking the logarithm of Equation (5.20) can be written as

$$l_{\boldsymbol{c}}(\boldsymbol{\Theta}|y,\boldsymbol{\omega},\boldsymbol{x}) = \sum_{i=1}^{N} z_{i1} \ln \pi_1 I_{y_i=0} + \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig} \Bigg[\ln \pi_g + \ln \text{Pois}\Big(y_i|\mu_{ig}(\boldsymbol{x}_i,\boldsymbol{\beta}_g)\Big) + \ln \phi\Big(\boldsymbol{q}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big) + \ln p\Big(\boldsymbol{w}_i;\boldsymbol{\alpha}_g\Big)\Bigg] \tag{5.21}$$

where $\pi_1 = 1 - \sum_{g=2}^{G} \pi_g$ and $\boldsymbol{\Theta} = (\boldsymbol{\mu}_g, \boldsymbol{\Sigma_g}, \boldsymbol{\beta}_g, \boldsymbol{\alpha}_g)$ for $g = 2, ...G$ is the set of model parameters to be estimated.

5.4.1 E-step

Using the current estimates $\boldsymbol{\Theta}^{(t)}$, we compute the probability $z_{ig}^{(t)}$ that the subject i comes from gth component of the mixture:

$$E(z_{ig}|y_i,\boldsymbol{\omega}_i,\mathbf{x}_i,\boldsymbol{\Theta}^{(t)}) = \frac{\pi_g \text{Pois}\Big(y_i|\mu_{ig}(\boldsymbol{x}_i,\boldsymbol{\beta}_g^{(t)})\Big)\phi\Big(\boldsymbol{q}_i|\boldsymbol{\mu}_g^{(t)},\boldsymbol{\Sigma}_g^{(t)}\Big)\text{Mult}\Big(\boldsymbol{w}_i,\boldsymbol{\alpha}_g^{(t)}\Big)}{p\Big(\boldsymbol{x}_i,y_i;\boldsymbol{\Theta}^{(t)}\Big)} \tag{5.22}$$

which correspond to the posterior probability that the unlabeled observation $(\boldsymbol{x}_i', y_i)'$ belong to the gth component of the mixture, using the current fit $\boldsymbol{\Theta}^{(t)}$ for $\boldsymbol{\Theta}$ and $z_{i1}^{(t)} = 1 - \sum_{g=2}^{G} z_{ig}^{(t)}$. The E-step, on the tth iteration, requires the calculation of

$$Q(\tau;\tau^{(t)}) = E_{\tau^{(t)}}[l_c(\tau)|\boldsymbol{\Theta}] \tag{5.23}$$

5.4.2 M-step

At this step, we obtain the $Q(.)$ with respect to $\boldsymbol{\Theta}^{(t+1)}$ where $t = 0, 1, \ldots,$ The conditional expectation of $l_c(\boldsymbol{\Theta})$ given the observed data , say $Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(t)})$ is maximized with respect to $\boldsymbol{\Theta}$. The z_{ig} are simply replaced by the current expectations $z_{ig}^{(t)}$

yielding,

$$Q\Big(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(t)}\Big) = \sum_{i=1}^{N} z_{i1}^{(t)} \ln \pi_1 I_{y_i=0} + \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig}^{(t)} \ln \pi_g + \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig}^{(t)} \ln p\Big(\boldsymbol{w}_i;\boldsymbol{\alpha}_g\Big)$$

$$+ \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig}^{(t)} \ln \text{Pois}\Big(y_i|\mu_{ig}(\boldsymbol{x}_i,\boldsymbol{\beta}_g)\Big) + \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig}^{(t)} \ln \left(\phi\Big(\boldsymbol{q}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right) \tag{5.24}$$

Substituting the values z_{ig} in Equation (5.21) with the values $z_{ig}^{(q)}$ obtained in Equation (5.22), we have

$$Q(\boldsymbol{\tau};\boldsymbol{\tau}^{(t)}) = Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(t)}) + Q_2(\boldsymbol{k};\boldsymbol{\tau}^{(t)}) + Q_3(\boldsymbol{\pi};\boldsymbol{\tau}^{(t)}) \tag{5.25}$$

where

$$Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(t)}) = \sum_{i=1}^{N}\sum_{g=2}^{G} z_{ig}^{(t)} \ln \left[\exp[-\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)]\frac{[-\mu_{ig}(\boldsymbol{x};\boldsymbol{\beta}_g)]^{y_i}}{y!}\right] \tag{5.26}$$

$$Q_2(\boldsymbol{\pi};\boldsymbol{\tau}^{(t)}) = \sum_{i=1}^{N} z_{i1} \ln \pi_1 I_{y_i=0} + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(q)} \ln \pi_g \tag{5.27}$$

$$Q_3(\boldsymbol{k};\boldsymbol{\tau}^{(t)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(t)} \ln \phi_d(\boldsymbol{q}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g) \tag{5.28}$$

$$Q_4(\boldsymbol{h};\boldsymbol{\tau}^{(t)}) = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}^{(t)} \ln \text{Mult}(\boldsymbol{w}_i;\boldsymbol{\alpha}_g) \tag{5.29}$$

5.4.3 Maximization of $Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(q)})$ via IRLS

We maximize Equation (5.23) independently of the G expression as the four terms on the right of Equation (5.23) have zero cross-derivatives, i.e.

$$\frac{\partial}{\partial\boldsymbol{\beta}_g} Q_1(\boldsymbol{\Omega};\boldsymbol{\tau}^{(t)}) = \frac{\partial}{\partial\boldsymbol{\beta}_g} \sum_{i=1}^{N} z_{ig}^{(t)} \ln \left(\text{Pois}\Big(y_i|\mu_{ig}(\boldsymbol{x}_i,\boldsymbol{\beta}_g)\Big)\right) \tag{5.30}$$

$$= \frac{\partial}{\partial\boldsymbol{\beta}_g} \sum_{i=1}^{n} z_{ig}^{(t)} \log \frac{\Big(\mu_{ig}(\boldsymbol{x}_i;\boldsymbol{\beta}_g)\Big)^{y_i} \exp\Big(-\mu_{ig}(\boldsymbol{x}_i;\boldsymbol{\beta}_g)\Big)}{y_i!} \tag{5.31}$$

$$= \frac{\partial}{\partial \boldsymbol{\beta}_g} \sum_{i=1}^{n} z_{ig}^{(t)} \Big[y_i \log \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) - \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) - \log y_i! \Big] \tag{5.32}$$

$$= \frac{\partial}{\partial \boldsymbol{\beta}_g} \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) = \boldsymbol{x}_i \exp(\boldsymbol{x}_i' \boldsymbol{\beta}_g) = \boldsymbol{x}_i \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) \tag{5.33}$$

$$= \frac{\partial}{\partial \boldsymbol{\beta}_g} \sum_{i=1}^{n} z_{ig}^{(t)} \Big[y_i \log \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) - \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) - \log y_i! \Big] \tag{5.34}$$

From Equation (5.34), we have the following

$$\sum_{i=1}^{n} z_{ig}^{(t)} \boldsymbol{x}_i \mu_{ig}\Big(\boldsymbol{x}_i; \boldsymbol{\beta}_g\Big) \left(\frac{y_i}{\mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)} - \frac{\mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)}{\mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)} \right) = 0 \tag{5.35}$$

$$S(\boldsymbol{\beta}_g^{(t)}) = \sum_{i=1}^{n} z_{ig}^{(t)} \boldsymbol{x}_i \Big(y_i - \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) \Big) \tag{5.36}$$

Using the expression $\boldsymbol{\beta}_g^{(t+1)} = \boldsymbol{\beta}_g^{(t)} + \Big[I(\boldsymbol{\beta}_g^{(t)}) \Big]^{-1} S(\boldsymbol{\beta}_g^{(t)})$, where $I(\boldsymbol{\beta}_g^{(t)})$ is the Fisher Information matrix and $S(\boldsymbol{\beta}_g^{(t)})$ is the score function obtained in Equation (5.36). So,

$$I(\boldsymbol{\beta}_g^{(t)}) = -\frac{\partial}{\partial} S(\boldsymbol{\beta}_g^{(t)}) = \sum_{i=1}^{n} z_{ig}^{(t)} \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) \boldsymbol{x}_i' \boldsymbol{x}_i \tag{5.37}$$

$$\boldsymbol{\beta}_g^{(t+1)} = \boldsymbol{\beta}_g^{(t)} + \left(\sum_{i=1}^{n} z_{ig}^{(t)} \mu_{ig}\Big(\boldsymbol{x}_i; \boldsymbol{\beta}_g\Big) \boldsymbol{x}_i' \boldsymbol{x}_i \right)^{-1} \left(\sum_{i=1}^{n} z_{ig}^{(t)} \boldsymbol{x}_i \Big(y_i - \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) \Big) \right) \tag{5.38}$$

finally Equation (5.38) becomes

$$\boldsymbol{\beta}_g^{(t+1)} = \left(\sum_{i=1}^{n} \boldsymbol{x}_i' s_{ig} \boldsymbol{x}_i \right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{x}_i' s_{ig} \delta_{ig}^{(t)} \right) \tag{5.39}$$

where $\delta^{(t)} = \boldsymbol{x}_i' \boldsymbol{\beta}_g + \delta_{ig}^*$ with $\delta_{ig}^* = \Big(y_i - \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g) \Big) \Big/ \mu_{ig}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)$ and $s_{ig} = z_{ig}^{(t)} \mu_{ig}^{(t)}(\boldsymbol{x}_i; \boldsymbol{\beta}_g)$

5.4.4 Maximization of $Q_2(\boldsymbol{\pi}; \boldsymbol{\tau}^{(t)})$ via ML

The maximum of $Q(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\pi}$, subject to the constraints on these parameters, $Q(\boldsymbol{\Theta})$ is achieved by maximizing the augmented function

$$= \sum_{i=1}^{N} z_{i1}^{(t)} \ln \pi_1 - \gamma_1 \Bigg(\pi_1 - 1 \Bigg) I_{y_i=0} + \sum_{i=1}^{N} \sum_{g=2}^{G} z_{ig}^{(t)} \ln \pi_g + \gamma \Bigg(\sum_{g=2}^{G} \pi_g - 1 \Bigg) \tag{5.40}$$

where γ_1 is a Lagrangian multiplier. Setting the derivative in Equation (5.40) with respect to π_1 for a degenerate distribution and π_g for a Poisson cluster weighted models equal to zero and solving yields

$$\pi_1^{(t+1)} = \sum_{i=1}^{n} z_{i1}^{(t+1)} \Bigg/ n$$

$$\pi_g^{(t+1)} = \sum_{i=1}^{n} z_{ig}^{(t+1)} \Bigg/ n \tag{5.41}$$

where $g = 2, \ldots, G$.

5.4.5 Maximization of $Q_3(\boldsymbol{k}; \boldsymbol{\tau}^{(t)})$ via ML

Maximizing Equation (5.23) with respect to $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$, $g = 1, ..., G$, is equivalent to maximizing independently each of the G expression

$$\sum_{i=1}^{n} z_{ig}^{(t)} \ln \Bigg(\phi\big(\boldsymbol{q}_i | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\big) \Bigg) \tag{5.42}$$

we obtain

$$\boldsymbol{\mu}_g^{(t+1)} = \sum_{i=1}^{n} z_{(ig)}^{(t)} \boldsymbol{q}_i \Bigg/ \sum_{i=1}^{n} z_{(ig)}^{(t)} \tag{5.43}$$

and

$$\boldsymbol{\Sigma}_g^{(t+1)} = \sum_{i=1}^{n} z_{ig}^{(t)} \big(\boldsymbol{q}_i - \boldsymbol{\mu}_g^{(t+1)}\big)\big(\boldsymbol{q}_i - \boldsymbol{\mu}_g^{(t+1)}\big)' \Bigg/ \sum_{i=1}^{n} z_{ig}^{(t)} \tag{5.44}$$

5.4.6 Maximization of $Q_4(\boldsymbol{h}; \boldsymbol{\tau}^{(t)})$ via ML

Maximizing $Q_4(\boldsymbol{h}; \boldsymbol{\tau}^{(t)})$ with respect to $\boldsymbol{\alpha}_g$, $g = 1, \ldots, G$ is equivalent to maximizing each of the G expression considering the constraints

$$\sum_{i=1}^{n} z_{ig}^{(t)} \ln p(\boldsymbol{w}_i; \boldsymbol{\alpha}_g) = \sum_{k=1}^{p} \sum_{i=1}^{n} z_{ig}^{(t)} \sum_{s=1}^{r_k} w^{ks} \ln \alpha_{gks}. \tag{5.45}$$

Using the Lagrangian multiplier, Equation (5.45) can be expanded as follows;

$$\sum_{i=1}^{n} z_{ig}^{(t)} \sum_{s=1}^{r_k} w^{ks} \ln \alpha_{gks} - \gamma_2 \left(\sum_{s=1}^{r_k} \alpha_{gks} - 1 \right), \tag{5.46}$$

the γ_2 is used here as the Lagrangian multiplier. Setting the derivative of Equation (5.46) to zero

$$\sum_{i=1}^{n} z_{ig}^{(t)} \frac{w^{ks}}{\alpha_{gks}} - \gamma_2 = 0, \implies \sum_{i=1}^{n} z_{ig}^{(t)} w^{ks} \Big/ \gamma_2 = \alpha_{gks}, \tag{5.47}$$

we find $\gamma_2 = \sum_{i=1}^{n} z_{ig}^{(t)}$, then finally we have

$$\alpha_{gks}^{(t+1)} = \sum_{i=1}^{n} z_{ig}^{(t)} w^{ks} \Big/ \sum_{i=1}^{n} z_{ig}^{(t)}. \tag{5.48}$$

We note here that we can also maximize $Q_4(\boldsymbol{h}; \boldsymbol{\tau}^{(t)})$ with respect to $\boldsymbol{\alpha}_g$ via IRLS if we allow probabilities $\boldsymbol{\alpha}_g$ to be either a multinomial logit or softmax regression with varying coefficients according to chapter 4.

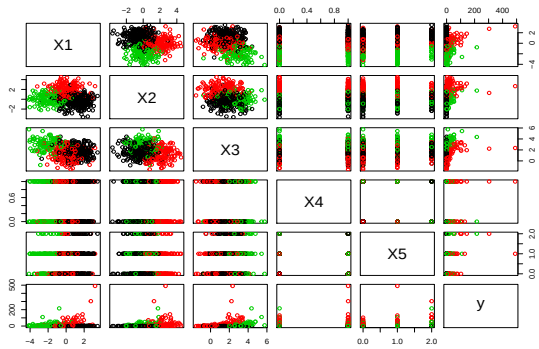


Figure 5.1: The Visualization of the simulated data with sample size 500

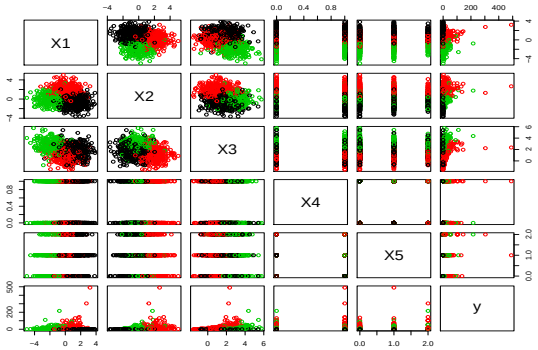


Figure 5.2: The Visualization of the simulated data with sample size 1000

5.5 A Simulation Study for parameter Recovery

A simulation study was performed to evaluate the performance of the maximum likelihood estimates of the model obtained via EM algorithm. We generated samples of size N from the following ZIPCWM with three components.

$$p(\boldsymbol{x}, y; \boldsymbol{\Theta}) = \pi_1 I_{(y=0)} + \sum_{g=2}^{3} Pois\Big(y\Big|\boldsymbol{x}; \boldsymbol{\beta}_g\Big)\phi\Big(\boldsymbol{q}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big)p\Big(\boldsymbol{w}; \boldsymbol{\alpha}_g\Big)\pi_g \tag{5.49}$$

Figure (5.1) and Figure (5.2) present the visualization of the simulated data with sample sizes 500 and 1000 but we intentionally not include visualization for $n = 200$. Table 5.1 shows the true values of the mean vectors, and the covariance matrices are the spherical covariance matrices. For the finite discrete variables w_1 and w_2, we generated two different variables from binomial distribution where $p = 2$ discrete covariates and $r_1 = 2$ and $r_2 = 3$ levels with the probability of success 0.5 and 1/3 respectively.

We randomly generated the continuous variables from a Gaussian variate according to the parameter presented in Table (5.1) whose dimension is $d = 3$. The two categorical variables $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are generated according to the binomial distribution where the probabilities are $\boldsymbol{\alpha_1} = 0.5$ with levels 2 and $\boldsymbol{\alpha}_2 = 1/3$ with levels 3 respectively. The

count response variable is generated according to the true values of the coefficients presented in Table (5.2).

Table 5.1: True values of the mean, sigma, and mixing weight for $n = 200, 500$ and 1000

	n	g	μ_1	μ_2	μ_3	σ_{11}	σ_{22}	σ_{33}	$\boldsymbol{\pi}$
True		1	-	—	—	—	-	-	0.50
		2	0.10	2.00	1.00	1.00	1.00	1.00	0.30
		3	−2.00	0.00	3.00	1.00	1.00	1.00	0.20

Table 5.2: True values of the regression coefficients for $n = 200, 500, 1000$

Values	n	g	β_0	β_1	β_2	β_3	β_4	β_5
True		2	0.00	0.88	0.28	0.96	0.09	0.33
		3	0.00	0.77	0.53	0.98	0.07	0.37

The log-link for the Poisson mean μ_{ig} is as follows:

$$\log(\mu_{i2}(\boldsymbol{x}_i, \beta_2)) = \beta_{20} + \beta_{21}\boldsymbol{x}_i$$

$$\log(\mu_{i3}(\boldsymbol{x}_i, \beta_3)) = \beta_{30} + \beta_{31}\boldsymbol{x}_i \tag{5.50}$$

where $\boldsymbol{x}_i$ in this case is the combination of $\boldsymbol{q}_i$, and $\boldsymbol{w}_i$ random variables. We considered the distribution $\mathbf{q}|\mathcal{D}_g \in \mathcal{R}^d$ to follow a Gaussian distribution. The true values of the ZIPCWM regression coefficients are presented in Table (5.2). We selected the $\pi_1 \approx 0.5$, $\pi_2 \approx 0.3$, and $\pi_3 \approx 0.2$ according to Hwa et al. (2014).

We generated a random number U from Uniform distribution. To generate samples from the above model for each subject $(i = 1, ..., n)$, we adopted the following conditions; if U is less than π_1, Y_i takes the value 0, where $\mu_{i1} = 0$. If U is between π_1 and $\pi_1 + \pi_2$, then Y_i is a draw from Pois(μ_{i2}). Otherwise, Y_i is generated from Pois(μ_{i3}). Algorithm 5 presents the steps taken to generate the simulated data;

5.5.1 Algorithm for simulating from ZIPCWM

Algorithm 5 Algorithm for simulation ZIPCWM

Select an initial coefficient $\boldsymbol{\beta}$, Number of Groups G
Initialize the group mean and Covariance matrix $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively,
Initialize assignment probability π_g
Set a seed
Generate $U \sim (0, 1)$
while $i \neq n$ **do**
  if $U_i \leq \pi_1$ **then**
    Generate $\boldsymbol{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$
    Generate $\boldsymbol{w1}_i \sim Bin(1, 0.5)$
    Generate $\boldsymbol{w2}_i \sim Bin(1, 1/3)$
    Combine the $\boldsymbol{x}_i = (\boldsymbol{q}_i, \boldsymbol{w}1_i, \boldsymbol{w}2_i)$
    Generate $\boldsymbol{y}_i \sim zipois(1, \mu_{i1} = 0)$
  else if $U_i \geq \pi_1$ & $U_i \leq \pi_1 + \pi_2$ **then**
    Generate $\boldsymbol{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
    Generate $\boldsymbol{w1}_i \sim Bin(1, 0.5)$
    Generate $\boldsymbol{w2}_i \sim Bin(1, 1/3)$
    Combine the $\boldsymbol{x}_i = (\boldsymbol{q}_i, \boldsymbol{w}1_i, \boldsymbol{w}2_i)$
    Generate $\boldsymbol{y}_i \sim \text{Pois}(1, \mu_{i2}(\boldsymbol{x}_i, \boldsymbol{\beta}_2))$
  else
    Generate $\boldsymbol{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$
    Generate $\boldsymbol{w1}_i \sim Bin(1, 0.5)$
    Generate $\boldsymbol{w2}_i \sim Bin(1, 1/3)$
    Combine the $\boldsymbol{x}_i = (\boldsymbol{q}_i, \boldsymbol{w}1_i, \boldsymbol{w}2_i)$
    Generate $\boldsymbol{y}_i \sim \text{Pois}(1, \mu_{i3}(\boldsymbol{x}_i, \boldsymbol{\beta}_3))$
  end if
end while

5.5.2 Result for Parameters Estimated

Here, the ZIPCWM with three component is fitted to the simulated data of sizes $N = 200, 500$, and 1000. We compute the misclassification based on the simulated data. We ran each simulation 10 times as suggested by McLachlan & Peel (2000) to avoid local maxima. However, we ran the simulation 10 times and recorded the average values to avoid the biased choice of selecting the best result.

Table (5.3) shows the recovered estimates of the $\boldsymbol{\mu}_g$, the diagonal values for $\boldsymbol{\Sigma}_g$, and the mixing proportions $\boldsymbol{\pi_g}$. For the estimated parameters of μ_1 in component 1 for $n = 200, 500$, and $n = 1000$, the values are omitted. Generally, in Table (5.3), we observe that the closeness of the parameter values is not independent of the sample size N. Also, in Table (5.4), considering the coefficients $\boldsymbol{\beta}_g$ where $g = 2, \ldots, G$, the parameter estimates for $N = 1000$ are closer to the true parameter values compared to the parameter estimates for $N = 200$ and $N = 500$.

Table 5.3: True and Recovered values of the mean, sigma, and mixing weight for $n = 200, 500$ and 1000

	N	g	μ_1	μ_2	μ_3	σ_{11}	σ_{22}	σ_{33}	$\boldsymbol{\pi}$
		1	-	–	–	–	-	-	0.50
True		2	0.10	2.00	1.00	1.00	1.00	1.00	0.30
		3	−2.00	0.00	3.00	1.00	1.00	1.00	0.20
		1	–	–	–	–	–	–	0.54
Recovered	200	2	−0.77	1.50	2.20	1.99	1.62	1.48	0.25
		3	−1.07	0.92	2.40	2.14	1.57	1.86	0.21
		1	–	–	–	–	–	–	0.53
	500	2	−0.50	1.52	2.03	1.72	1.39	1.25	0.25
		3	−1.44	0.62	2.79	1.45	1.30	0.99	0.22
		1	–	–	–	–	–	–	0.53
	1000	2	0.17	2.08	1.17	0.70	0.81	1.11	0.27
		3	−1.93	−0.07	3.09	1.00	1.05	1.21	0.20

Table 5.4: True and Recovered values of the regression coefficients for $n = 200, 500, 1000$

	N	g	β_0	β_1	β_2	β_3	β_4	β_5
True		2	0.00	0.88	0.28	0.96	0.09	0.33
		3	0.00	0.77	0.53	0.98	0.07	0.37
Recovered	200	2	0.33	0.73	0.26	0.87	0.07	0.39
		3	0.30	0.71	0.29	0.84	0.18	0.41
	500	2	0.25	0.82	0.29	0.92	0.09	0.32
		3	0.24	0.75	0.42	0.89	0.086	0.38
	1000	2	0.02	0.92	0.27	0.86	0.09	0.39
		3	0.51	0.81	0.49	0.86	0.04	0.41

We investigate the ability of the proposed model to identify the number of components using eight different model selection criteria such as Akaike's information criterion (AIC) and Bayesian information criterion (BIC) model selection criteria, ICL, AWE, AICc, AIC3, AICu, Caic.

The values presented are the average values of ten runs for each of the criterion. In Table (5.5), the effect of sample size is significantly evident. It can be seen that all the selection criteria select wrong number of components with large size. They all selected the model with too many components (clusters) when the sample size is $N = 200$. However, when the sample size increases to 500, AIC, AIC3, and AICc correctly selected the true cluster component. Moreover, when the sample size is 1000, all the selection criteria except AWE selected the right choice of the model component. AWE displays a poor performance of overestimating the number of components in the data. All criteria except the AWE performed satisfactorily with higher sample size. Obviously, we can conclude that the performance of the criteria roughly gets better with increasing sample size N.

Table 5.5: The values of AIC, BIC, ICL of ZIPCWM for different $n = 200, 500, 1000$ with true component $G = 3$. For $n = 200$, all the selection criteria perform poorly. However, the performance gets improved with increased sample size.

N	G	AIC	BIC	ICL	AWE	AIC3	AICc	AICu	Caic
200	2	2192.91	2080.77	2080.76	1798.62	2158.91	2178.48	2140.01	2046.77
	3	2055.82	1831.49	1822.53	1267.24	1987.82	1984.18	1899.56	1763.53
	4	1945.12	1608.69	1598.41	762.26	1843.12	1728.50	1583.78	1506.69
	5	**1771.85**	**1323.28**	**1315.37**	**194.70**	**1635.85**	**1180.35**	**949.32**	**1187.28**
500	2	7557.49	7414.20	7413.55	7100.90	7523.49	7552.37	7516.09	7380.20
	3	**5128.59**	4841.99	4826.93	4215.40	**5060.59**	**5106.82**	5032.57	4773.99
	4	5212.08	4782.19	4750.51	3842.30	5110.08	5159.16	5043.82	4680.19
	5	5270.85	**4697.66**	**4649.80**	**3444.47**	5134.85	5168.19	**5008.09**	**4561.66**
1000	2	12074.24	11907.38	11907.37	11570.51	12040.24	12071.78	12036.15	11873.38
	3	**10234.37**	**9863.09**	**9900.64**	9226.91	**10166.37**	**10224.29**	**10152.79**	**9832.64**
	4	10860.90	10360.31	10268.41	9349.72	10758.90	10837.48	10728.78	10258.31
	5	10907.28	10239.83	10127.11	**8892.37**	10771.28	10864.10	10716.76	10103.83

We presented the confusion matrix for ZIPCWM, PCWM [Ingrassia et al. (2015)], and FZIP [Hwa et al. (2014)] with varying sample sizes. Table (5.6) shows the misclassification rates produced by the competing models. The overall misclassification rate produced by ZIPCWM is 5.5%, 8.4%, and 6.7% for $N = 200$, 500, and 1000 respectively. Moreover for $N = 1000$, we compared proposed model with PCWM and FZIP. PCWM has a misclassification rate of 10.2 and FZIP has 13.1% misclassification rate. The classification power of FZIP agrees with the simulated study provided in Hwa et al. (2014). PCWM achieves a slightly higher classification accuracy of 89.8% compared to FZIP. To validate the classification power, we presented the ARI value for each model which measures the agreement between the true cluster and classification result of the proposed model. The ARI of the ZIPCWM is 0.892 when $N = 1000$ and FZIP has ARI of 0.847. This means the predicted result of ZIPCWM agrees more with the true classification than the FZIP. The AUC for both sample sizes are 0.923, while the AUC of FZIP model is 0.829.

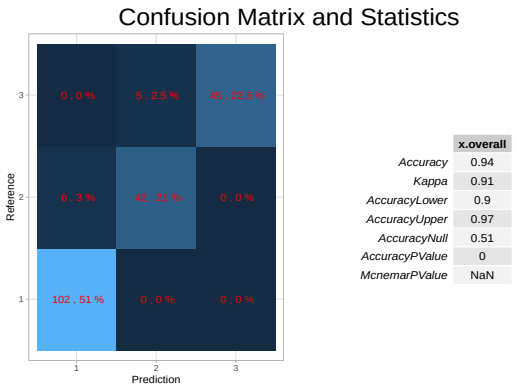


Figure 5.3: The Visualization of Confusion Matrix and Statistics of the ZIPCWM with $n = 200$, $G = 3$.

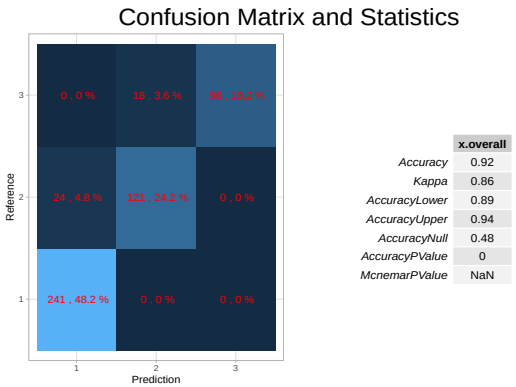


Figure 5.4: The Visualization of Confusion Matrix and Statistics of the ZIPCWM with $N = 500$, $G = 3$.

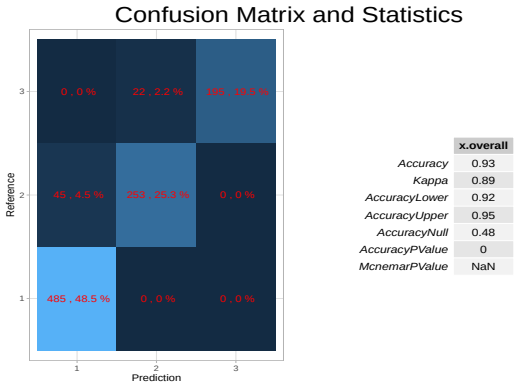


Figure 5.5: The Visualization of Confusion Matrix and Statistics of the ZIPCWM with $N = 1000$, $G = 3$.

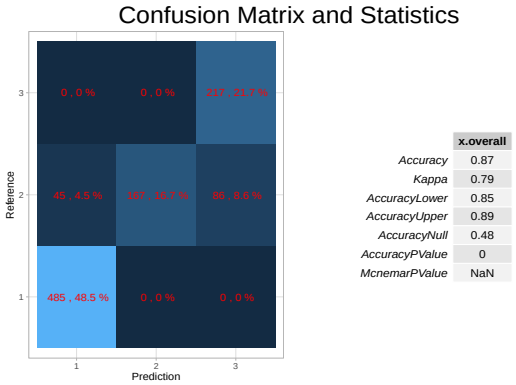


Figure 5.6: The Visualization of Confusion Matrix and Statistics of the FZIP regression mixture model with $N = 1000$, $G = 3$.

Table 5.6: Confusion Matrix in the three component Model for $N = 200, 500, 1000$. For 1000, ZIPCWM, PCWM, and FZIP are compared together. It was observed that ZIPCWM outperforms PCWM and FZIP.

n	Component	1	2	3	MR (%)
200	1	102	0	0	0.00
	2	6	42	0	12.50
	3	0	5	45	10.00
Misclassification	5.50				
Accuracy	94.50%				
500	1	241	0	0	0.00
	2	24	121	0	16.55
	3	0	18	96	15.79
rate	8.40				
Accuracy	91.60%				
1000	1	485	0	0	0.00
	2	45	253	0	15.10
	3	0	22	195	10.14
Misclassification	6.70				
Accuracy	93.30%				
PCWM	1	485	0	0	0.00
	2	15	196	87	34.23
	3	0	0	217	0.00
Overall	10.20				
ccuracy	89.80%				
FZIP	1	485	0	0	0.00
	2	45	167	86	43.96
	3	0	0	217	0.00
Overall	13.10				
ccuracy	86.90%				

5.6 Modeling of ZIPCWM on Real Data

5.6.1 The Use of Contraceptive Among married women

We revisited the data set on the use of contraceptive methods among married woman whose attributes have been presented in Chapter 4. The main goal is to first check if there is any presence of excess zeros or class imbalance in the data before classifying

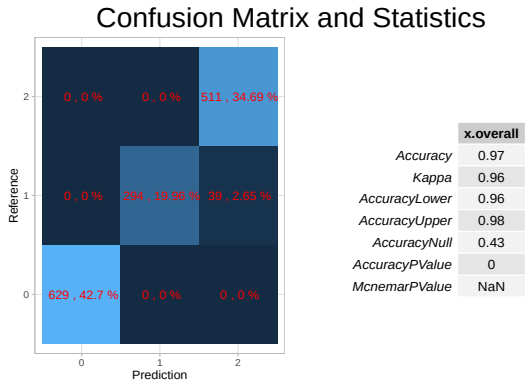


Figure 5.7: The Visualization of Confusion Matrix and Statistics of the ZIPCWM on Contraceptive data

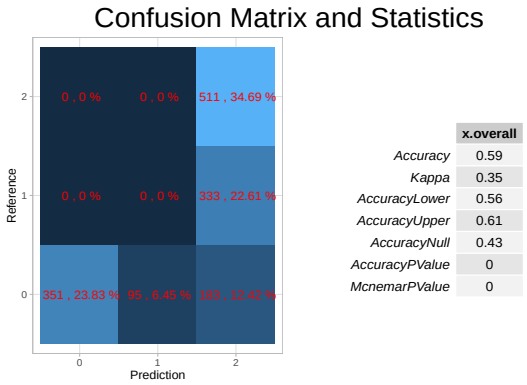


Figure 5.8: The Visualization of Confusion Matrix and Statistics of the PCWM on Contraceptive data

the current contraceptive method choices (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

First of, we carried out a dispersion test to ascertain the absence of overdispersion in the data. The dispersion parameter is 0.83983 which signifies the absence of overdispersion. Furthermore, the Poisson regression model was carried out to check the significant power of the variables in the data. The result shows that the wife's age has a significant effect on the use of contraception among married women. This is reasonable since the estimate $\beta_{\text{age}} = -0.04375$. This means that younger married women are 0.04375 times more expected to use contraceptive than their older married counterparts. More conceptually, the use of contraceptive method has about 4.4% decrease among married women for every one-year increase in age. Similarly, the wife's education also contributes significantly to the use of contraceptive with a *p-value* of $< 2e - 16$. The coefficient of education variable (1.18) indicates that on average there is an increase of 1.18 use of contraceptive method for every one-year education. This means that well-educated married women have more exposure to the use of contraceptive. The rest of the significant variables considered in the study are Number of children ever born, Standard-of-living index and Media exposure.
We performed a comparison study of three different models such as Zero-Inflated Poisson Cluster-Weighted Models (ZIPCWM), Poisson Cluster-weighted Models (PCWM, Ingrassia et al. (2015)), and Fixed Zero-Inflated Poisson Mixture models (FZIP, Hwa et al. (2014)) based on these significant variables.

Table 5.7: Comparison of classification performance of ZIPCWM and PCWM with classification accuracy of 97%, and 58.5% respectively. FZIP performs so poorly on the data as the model could not identify the reasonable component for the data.

Model	Component	0	1	2	MR (%)
ZIPCWM	0	629	0	0	0.00
	1	0	294	39	11.71
	3	0	0	511	0.00
Misclassification	2.60				
Accuracy	97.40%				
PCWM	0	351	95	183	44.20
	1	0	0	333	100.00
	2	0	0	511	0.00
Misclassification	41.50				
Accuracy	58.50%				

Here, the ZIPCWM, PCWM, and FZIP models with three components are fitted to the method of contraceptive among the married women based on the significant demographics and they are used to classify the data into three components. We computed the misclassification rate produced by the competing models. Visualization results are presented in Figure (5.7) and Figure (5.8). In Table (5.7), ZIPCWM has the overall misclassification rate is 2.6%. Moreover, it can be seen that most of the misclassifications are in component two which has 11.71%. In group one the number of married women according to the prediction of the model is 629. Contrary to the result produced in Chapter 4, ZIPCWM appropriately accounts for the excess zeros in the data. The component two according to ZIPCWM has 294 married women correctly classified as the married women that have long-term use of contraceptive. 39 observations of the women that have long-term use were misclassified into cluster three (short-term use). ZIPCWM correctly predicts that 511 married women have a short-term use of contraceptive. We observe that PCWM has a higher overall misclassification rate of 41.50%. PCWM correctly classified 351 out of 629 married women as a no-use contraceptive method. This is due to the PCWM not accounting for the excess zeros in the data. Moreover, all the component two was totally misclassified as component three. This is a problem of label switching. FZIP provides the worst classification. FZIP was unable to identify the components at all but classified all the women as coming from component three. We conclude based on the result produced

by the competing models that ZIPCWM appropriately fit better than PCWM and FZIP for the use of contraceptive method among married women.

5.6.2 Modeling the Number of Absence of students

The data consists of the performance of students in Portugal. The secondary education consists of 3 years of schooling, preceding 9 years of basic education followed by the higher education. Due to some factors, most of the students join the public and free education system. The data was collected primarily with questionnaires. The structure of the education system in Portugal is a 20-point grading scale where 0 is the lowest and 20 is the perfect grade. The data was collected during the $2005-2006$ school year from two public schools by Cortez & Silva (2008). They designed the latter with closed questions related to demographic, social/emotional [Pritchard & Wilson (2003)] and school related questions. The goal of the study is to classify the students according to the number of absence in the class. This is one major cause of zero inflation in the data if there is no significant reason for any student to be absent in class, the number of absence will be greatly zeros.

First of, we identify the significant factors that contribute to the absence. We modeled the data with a Poisson regression to investigate the significant effect of the variables on the presence of the students in class. Table (5.8) shows the significant variables that contributed to the absence/presence of the students in class. It is interesting to know how some factors can strongly contribute to or negatively affect the success of the students such as parent cohabitation status. The response variable is taken to be a Poisson count. We also note the presence of class imbalance in the data i.e. one class makes up about 63% of the response variable which makes it suitable for Zero-inflated model.

The response variable is a number of event occurring (absence) in the class. The number of absence ranges from 0 to 93 where 0 represents no absence and other numbers represent the number of absence. The response variable can also be seen as a binary variable of presence and absence. However, we performed a model selection test using the combination of eight model selection criteria with ZIPCWM and the result is presented below.

Table 5.8: The significant student related variables

Attribute	Description
school	student's school (binary: *Gabriel Pereira* or *Mousinho da Silveira*)
sex	student's sex (binary: female or male)
age	student's age (numeric: from 15 to 22)
Pstatus	parent's cohabitation status (binary: urban or rural)
Medu	mother's education (numeric: 0 to 4)
Fedu	father's education (numeric: 0 to 4)
Mjob	mother's job (nominal)
Fjob	father's job (nominal)
guardian	student's guardian (nominal: "mother", "father" or "other")
studytime	weekly study time
failures	number of past class failures (numeric: n if $1 \leq n < 3$, else 4)
famsize	family size (binary: ≤ 3 or > 3)
schoolsup	extra educational support (binary: yes or no)
famsup	family educational support (binary: yes or no)
paidclass	extra paid classes(binary: yes or no)
higher	wants to take higher education? (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
freetime	free time after school (numeric: from 1 - very low to 5 - very high)
Dalc	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
health	current health status (numeric: from 1 - very bad to 5 - very good)
absences	number of school absences (numeric: from 0 to 93)

Figure (5.9) and Figure (5.10) present the visualization of the confusion matrix with the classification accuracy. Thus, we conclude that ZIPCWM has a higher classification power compared to PCWM. We measure the agreement between actual class and predicted class using Adjusted Rand Index (ARI). ZIPCWM has ARI of 1 while PCWM has ARI of 0.81. After confirming the component of the model, we compared the classification power of ZIPCWM and PCWM models to distinguish between the zero-inflation model and ordinary model. We observed that the ZIPCWM outperforms PCWM.

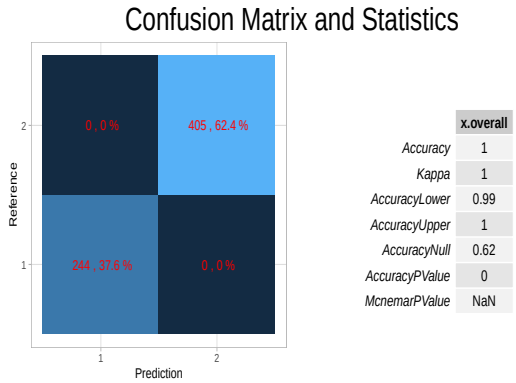


Figure 5.9: The Visualization of Confusion Matrix and Statistics of the ZIPCWM on number of Absence.

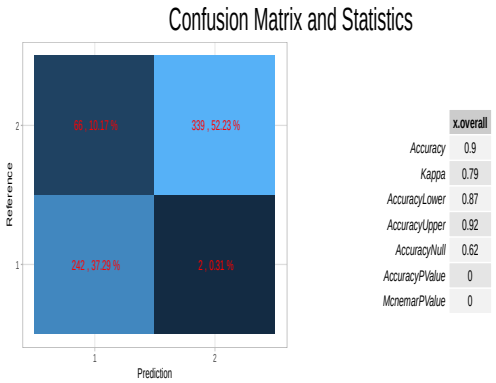


Figure 5.10: The Visualization of Confusion Matrix and Statistics of the PCWM on number of Absence

Table (5.9) confirms our intuition about the response variable. We observe that all the model selection criteria agree with the selection of the model with two components. This shows us that there are two categories of students in the class viz; those students that are regular in class and those that are irregular whether short-term or long-term.

Table 5.9: The values of model selection criteria of ZIPCWM for different $G = 2, 3, 4$ and 5. Before proceeding with the clustering analysis, we carried out a model selection test. All the selection criteria used suggest the model with two components.

G	AIC	BIC	ICL	AWE	AIC3	AICc	AICu	Caic
2	**6505.81**	**6420.77**	**6419.84**	**6240.74**	**6486.81**	**6504.60**	**6484.28**	**6401.77**
3	7784.37	7614.30	7481.70	7254.24	7746.37	7779.51	7739.29	7576.30
4	8059.74	7804.64	7633.21	7264.54	8002.74	8048.55	7987.79	7747.64
5	7515.60	7175.47	7062.70	6455.34	7439.60	7495.14	7413.18	7099.49

This ascertain a need to account for excess zeros in the data. Table (5.10) shows the misclassification rate of both models. ZIPCWM has zero misclassification rate and achieves 100% classification accuracy while the PCWM has 89.52% classification accuracy with 10.48% error of classification.

Table 5.10: Comparison of classification power of the ZIPCWM and PCWM Models. We intentionally omitted FZIP because we are interested in zero-inflated model against ordinary model. ZIPCWM provides higher classification accuracy relative to PCWM.

Model	Component	1	2	MR (%)
ZIPCWM	1	244	0	0.00
	2	0	405	0.00
Misclassification		0.00		
Accuracy		100.00%		
PCWM	1	242	2	0.82
	2	66	339	16.30
Misclassification		10.48		
Accuracy		89.52%		

5.7 Summary

In this Chapter, we have introduced another new member called Zero-Inflated Poisson cluster-weighted model (ZIPCWM) into the family of CWMs which accounts for excess zeros in the data. The ZIPCWM is suitable for class imbalance in data commonly known as censored information in medical data. ZIPCWM is a generalized form of the previously existing models such as Poisson cluster weighted model, Fixed Zero-Inflated Poisson mixture model. ZIPCWM allows modeling the data with mixed-type covariates which is the combination of the finite discrete and continuous variables.

This provides an advantage over the limitations of GZIP and FZIP. Furthermore, we have extensively described an Expectation-Maximization algorithm for parameter estimation via IRLS. To investigate the parameter recovery of the algorithm and the performance of various model selection criteria to select the number of mixture components, explicit simulation studies were carried out.
Our simulation showed that the proposed model worked satisfactorily and the estimation technique performed well. The results presented showed that the ZIPCWM with three components provides the best fit. This results lent support to the use of cluster weighted model and establish that ZIPCWM provides better fitting performance than the PCWM and FZIP when explaining zero-inflated heterogeneous data. Addition-

ally, the proposed model was applied to the real data and a comparison study of the classification performance with its special models was investigated. It was observed that the proposed model outperformed its special models. In this Chapter, we used eight different model selection criteria to select the number of mixing components, G. This is necessary to discover the hidden structure of the data irrespective of the response variable. The future research direction can be to critically investigate the eigenvalue decomposition on ZIPCWM.

The next chapter was carried out with Prof. Robert Aykroyd in the University of Leeds on image recognition. The chapter focuses on the problems encountered in Expectation propagation algorithm when in dealing with complex Bayesian hierarchical model. The chapter aims to solve some of the problems by proposing a stochastic and deterministic generalization of EP.

Chapter 6

Variational Bayesian: Expectation Propagation Image Reconstruction

6.1 Introduction

Image reconstruction problems common to medical imaging area such as fast MRI and low dose CT are generally mathematically ill-posed inverse problems. Often times, linear imaging system are considered with a forward operator G, e.g. a Fourier transform for MRI and X-ray transform for CT. The measurement y is given as $y = Gx$, where x is the underlying image in the noise-free state. The linear operator G is ill-posed for most applications; therefore, some statistical priors are necessary to make these problems invertible. Consequently, intractability becomes inevitable due to high-dimensional data. Moreover, when the number of hidden variables to be estimated grows bigger and result in multiple integration, intractability poses a problem in Bayesian inference.

This Chapter presents a new unification of stochastic algorithm and deterministic algorithm called *Splitting Expectation Propagation* (SEP). A splitting expectation propagation achieves a very high accuracy and also in terms of computational speed is faster than the Markov Chain Monte Carlo method and other existing deterministic approximation methods like Assumed Density Filtering (ADF) and standard EP algorithms. Although, EP algorithms may not guarantee the desired result due to their limitations which will be discussed in the following paragraph. As pointed out

by Minka (2001); ADF is a one pass, sequential method for computing an approximate posterior distribution. The main advantage of ADF over EP is its simplicity and high memory efficiency. However, the weakness of ADF stems from its sequential nature, its sensitivity to observation ordering makes it undesirable in a batch context. Expectation Propagation [Minka (2001)] on the other hand is an extension of ADF [Maybeck (1982); Opper & Winther (1999)]. EP incorporates the term *iterative refinement*. The word iterative refinement uses an additional passes through the network. The choice made earlier is independent of the information from the latter observation but rather refines the choice which is important to retain. This iterative refinement poses a lot of problem to EP and often leads to algorithm explosion due to a negative value of the variance v_θ. This happens when many of the variances v_i of the approximate terms are negative and the positive v_i is to be refined. In the refinement stage, a positive value is subtracted from new approximate posterior variance v_θ^{new} and are left with some negative values. In this case, the marginal likelihood Z_i doesn't exist any longer and the algorithm fails. Although, a remedy to this was given in Minka (2001); which relaxes the expectation constraints: "set variance v_i to a large value (10^8) anytime it would become negative". This remedy leads to convergence but at the expense of the posterior accuracy. EP proves to be very difficult in handling hierarchical models due to the computation of marginal likelihood (Z_i) which may come from a different exponential family and yet does not guarantee an accurate result of the posterior. When the posterior is multimodal, EP fails to capture the whole mode because it is a unimodal approximate algorithm. An example is seen in clutter problem [Minka (2001)].

6.2 Related Work

Many attempts have been done on expectation propagation to solve the problem of instability. Quite a few researches have focused on the problem of energy function of expectation propagation i.e. the major part of the algorithm which is the minimization of KL divergence. Some have shifted their attention on the need to parallelize expectation propagation to reduce its inefficiency or ill-management of memory. Moreover, a lot of works have focused on applying the algorithm to various

areas of research. One application of expectation propagation has been found in the area of nonlinear inverse problem.

Matthias & Bangti (2014) studied expectation propagation method and demonstrated its significant potentials on the nonlinear inverse problem. To handle non-linearity in inverse problem, they proposed the coupling of the EP with an iterative linearization strategy. A direct integral was used for computing the marginal likelihood. This can only be tractable in a low dimensional space as the work assumed. However in practice, this becomes a hard nut to crack in hierarchical Bayesian framework. Also, their approach only involves the Gaussian distribution. Jose & Ryan (2015) worked on a similar problem handled in this work as a layered Bayesian Neural Network. Due to the memory inefficiency of EP and the possibility of the likelihood to grow with massive data, the method of EP was approached as an ADF in multiple succession. However, the disadvantage of this approach is that it can lead to underestimation of variance of approximate posterior. Graves (2011) proposed a Monte Carlo approximation to compute the lower bound of the variational inference, which is then optimized using the second approximation for the stochastic gradient descent (SDG). Following this approach for variational inference, the initial approximation leads to poor estimation for large data sets because of its inefficient use of data. John et al. (2011) focused on the problem of intractable lower bound on the marginal likelihood in variational inference while computing the updates of the parameters. It often requires the ability to integrate a sum of terms in the log joint likelihood using factorized distribution. However, not all the integrals are tractable which is typically handled by using a factorized distribution. In order to overcome this type of problem in variational inference, they presented an alternative approach based on stochastic optimization that allows for direct optimization of the variational lower bound.

Gelman et al. (2014) revisited expectation propagation as a prototype for scalable algorithm that partition big datasets into many parts and analyze each part in parallel to perform inference of shared parameters. The limitations of expectation propagation were highlighted and discussed in detail in their work. Jylanki et al. (2011) discovered that when the moment computations are not accurate, EP may have stability issues, even with one-dimensional tilted distributions, moment computations are more challenging if the tilted distribution is multimodal or has long tails. Minka

(2004) proposed an extension of EP called a fractional EP to improve the robustness of EP algorithm when the approximation family is not enough [Minka (2005)] or when the propagation of information is difficult due to vague prior information [Seeger (2008)]. Fractional EP can be viewed as a method for minimizing the α, where $\alpha = 1$ corresponds to Kullback-Leibler divergence used in EP, $\alpha = 0$ corresponds to the reverse Kullback-Leibler divergence used in variational Bayes, and $\alpha = 0.5$ corresponds to Hellinger distance. Yingzhen et al. (2018) presented an extension of expectation propagation called a Stochastic Expectation Propagation that maintains a global posterior approximation like variational Bayes but updates the posterior in a local way. The work used the expectation propagation strategy but differs only at the updating stage. Stochastic Expectation Propagation was seen as a corrected version of ADF such that it updates the global factor that captures the average effect of likelihood on the posterior. In this work, we study the complex Bayesian model by proposing a Splitting EP algorithm for image reconstruction.

6.2.1 Main contribution

The goal of this present work is to study the Expectation Propagation method, and to make a new modification of EP fit for the hierarchical Bayesian framework to solve an inverse problems. Splitting EP (SEP) incorporates the Monte Carlo methods (Stochastic Search), Markov Chain Monte Carlo method (MCMC), and Alternating Direction Method of Multiplier (ADMM) at the EP updating stage. SEP focuses on the core limitation of expectation propagation algorithm. The major limitation addressed in this work that automatically addresses other limitations is the computation of marginal likelihood in the hierarchical Bayesian models and also in high dimensional space. Recall that in the expectation propagation, the most vital part of the algorithm is the minimization of the Kullback-Leibler divergence between the tilted posterior and the approximate posterior. For background knowledge on EP, Minka (2001) provides detailed explanation. The new approach to the EP method is called the Splitting Expectation Propagation (SEP) algorithm. One of the advantages of Splitting Expectation Propagation is its ability to generalize expectation propagation when normalizing factor involves different exponential family of distributions (non-Gaussian distributions and Gaussian distributions).

Because of the intractability of inversion of an ill-posed operator with partial and corrupted measurements, we do not intend to learn an end-to-end inversion mapping from the measurements to the reconstructed image. Motivated by Bayesian inference-based image reconstruction methods, we propose to split the task of inversion of a known forward operator from learning an image representation. In order to feed the inputs into the EP algorithm implicitly, we approach EP from both stochastic and deterministic standpoints. From the stochastic viewpoint, we apply MCMC and Monte Carlo integration to EP setup and from the deterministic viewpoint we establish the normalizing factor as a loss function and apply the ADMM (EP-ADMM). Finally, we present EP with ADMM and MCMC (Combination of stochastic and deterministic standpoints to EP), and Stochastic Search EP independently (SSEP is the only Stochastic standpoint to EP) addresses the intractable normalizing factor of EP with Monte Carlo integration and gradient descent. This is altogether called *Splitting Expectation Propagation* (SEP).

6.3 Stochastic and Deterministic Methods

6.3.1 Markov Chain Monte Carlo

In Bayesian modeling, we encode our knowledge about the unknown x in a prior distribution $p(x)$ that forms the second building block of Bayesian modeling. The prior plays a role of regularization function of the ill-posedness pertinent to the model. The most widely used Bayesian algorithm is Markov Chain Monte Carlo (MCMC) [see Gamerman & Lopes (2006); Geyer (2011); Gilks (1995)], and it is a popular method for exploring posterior state. MCMC constructs a Markov chain with the posterior distribution $p(\mathcal{X}, \lambda, \tau|\mathcal{Y})$ as its stationary distribution, and draws samples from the posterior distribution by running the Markov chain, from which the sample mean and variance can be computed. Specifically, MCMC algorithm generates a set of N dependent samples which is used to approximate the posterior. In order to obtain accurate estimate a moderately large number of samples after burn-in period is needed for stationarity purpose which is controlled by the Autocorrelation of the samples, e.g., $N = 1 \times 10^5$ to 1×10^6. Each iteration is a decision stage of whether the proposal $x^{(i)}$ should be accepted or rejected. The major goal of this approach is

to evaluate the likelihood and the prior distributions. However, MCMC algorithm is extremely expensive and thus a straightforward application is impractical. The MCMC method provide an approach when the model is complex or the number of parameters is large. The transitions in the Markov chain are designed so that an equilibrium distribution exists and is equal to the target distribution, for example the posterior distribution in Bayesian analysis.

6.4 The Clutter Problem

The clutter problem has been addressed in the work by Minka (2001). This work replicates this process to compare the proposed algorithm with the EP algorithm for a one-dimensional space. Suppose we have observations from a Gaussian distribution embedded in a sea of unrelated clutter where w is the clutter ratio, so that the density observation is a mixture of two Gaussians:

$$p(\mathbf{y}|\theta) = (1-w)\mathcal{N}(\mathbf{y};\theta,\mathbf{I}) + w\mathcal{N}(\mathbf{y};\mathbf{0},10\mathbf{I}) \tag{6.1}$$

Let the d-dimensional vector $\boldsymbol{\theta}$ with a Gaussian prior distribution:

$$p(\theta) \sim \mathcal{N}(\mathbf{0},100\mathbf{I}) \tag{6.2}$$

The joint distribution of $\boldsymbol{\theta}$ and n observation $D = \{\mathbf{y_1},...,\mathbf{y_n}\}$

$$p(D,\theta) = p(\theta)\prod_i p(\mathbf{y}_i|\theta) \tag{6.3}$$

The goal is to approximate the posterior distribution

$$p(\boldsymbol{\theta}|D) = \frac{p(D,\theta)}{\int p(D,\theta)\,d\theta} \tag{6.4}$$

Here, the first component contain the parameter to be estimated which is θ and the second component describes the clutter where w is the known ratio of the clutter. The Bayesian network for this problem is θ point to the y_i [Check Minka (2001) for more detail]. The next subsection reveals the results of expectation propagation and splitting expectation propagation with a clutter problem in one-dimensional space.

6.4.1 SSEP in low-dimensional space

In order to address the most difficult but vital part of expectation propagation algorithm, we next present a method based on stochastic search and gradient descent for the evidence of the posterior distribution which involves integral in case of intractability. This method uses a stochastic approximation of the gradient with respect to the approximation distribution q. This is what we call SSEP.

Following is the definition of terms in EP algorithm

- $\tilde{t}_i(\theta)$ is the approximate term
- ω is the hyperparameter.
- $q(\theta)$ is the approximate posterior
- $q_{-i}(\theta)$ is the cavity distribution
- $\boldsymbol{m}_\theta$ is the mean of the approximate posterior
- v_θ is the variance of the approximate posterior
- $\tilde{p}_i(\theta)$ is the tilted distribution.

The Monte Carlo integration and gradient descent of the $\nabla_\omega \log Z_i(\theta|\omega)$ for each observation become

$$\nabla_\omega \log Z_i(\theta|\omega) = \nabla_\omega \log \int t_i(\theta)\, q_{-i}(\theta|\omega)\, d\theta \tag{6.5}$$

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int \nabla_\omega\, t_i(\theta)\, q_{-i}(\theta|\omega)\, d\theta \tag{6.6}$$

where $Z_i(\theta|\omega) = \int t_i(\theta)\, q_{-i}(\theta|\omega)\, d\theta$.
We substitute $q_{-i}(\theta|\omega) = \frac{q(\theta|\omega)}{\tilde{t}_i(\theta|\tilde{\omega})}$ in Equation (6.6) and it becomes

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int \nabla_\omega\, t_i(\theta)\, \frac{q(\theta|\omega)}{\tilde{t}_i(\theta|\tilde{\omega})}\, d\theta \tag{6.7}$$

we take the gradient with respect to the parameter of $q(\theta|\omega)$

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int \frac{t_i(\theta)}{\tilde{t}_i(\theta|\omega)} \, q(\theta|\omega) \, \nabla_\omega \log q(\theta|\omega) \, d\theta \tag{6.8}$$

We use $\nabla_\omega q(\theta|\omega) = q(\theta|\omega) \nabla_\omega \log q(\theta|\omega)$ in Equation 6.8

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int t_i(\theta) \, q_{-i}(\theta|\omega) \, \nabla_\omega \log q(\theta|\omega) \, d\theta \tag{6.9}$$

Equation (6.9) can be viewed in different ways; either by generating samples from the Uniform distribution;

$$\nabla_\omega \log Z_i(\theta|\omega) \approx \frac{1}{K} \sum_{k=1}^{K} t_i(\theta^{(k)}) \, q_{-i}(\theta^{(k)}|\omega) \, \nabla_\omega \log q(\theta^{(k)}|\omega) \Big/ \sum_{k=1}^{K} t_i(\theta^{(k)}) \, q_{-i}(\theta^{(k)}|\omega_{-i}) \tag{6.10}$$

or by rewriting it as a function of $q(\theta|\omega)$ and generating samples from it.

$$\nabla_\omega \log Z_i(\theta|\omega) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{t_i(\theta^{(k)})}{\tilde{t}_i(\theta^{(k)}|\tilde{\omega})} \, \nabla_\omega \log q(\theta^{(k)}|\omega) \Big/ \sum_{k=1}^{K} \frac{t_i(\theta^{(k)})}{\tilde{t}_i(\theta^{(k)}|\tilde{\omega})} \tag{6.11}$$

where $\theta^{(k)} \overset{iid}{\sim} q(\theta|\omega)$ for $k = 1, ..., K$. Equation (6.10) and Equation (6.11) can be denoted as δ_1/δ_0 so at iteration t,

$$\omega^{t+1} = \omega^t + lr * \frac{\delta_1}{\delta_0} \tag{6.12}$$

An alternative version of SSEP is to use a stochastic approximation of the gradient with respect to the old approximation posterior q_{-1}. The evidence or normalizing factor is defined

$$Z_i(\theta|\omega) = \int t_i(\theta) \, q_{-i}(\theta|\omega) \, d\theta \tag{6.13}$$

The Monte Carlo integration and gradient descent of the $\nabla_\omega \log Z_i(\theta|\omega)$ for each observation becomes

$$\nabla_\omega \log Z_i(\theta|\omega) = \nabla_\omega \log \int t_i(\theta)\; q_{-i}(\theta|\omega)\; d\theta \tag{6.14}$$

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int \nabla_\omega\; t_i(\theta)\; q_{-i}(\theta|\omega)\; d\theta \tag{6.15}$$

Our goal is to make a stochastic approximation of this gradient. To preserve the originality of the EP, we rewrite the function as

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int\; t_i(\theta)\; q_{-i}(\theta|\omega)\; \nabla_\omega \log q_{-i}(\theta|\omega) d\theta \tag{6.16}$$

We take the gradient with respect to the parameter of $q_{-i}(\theta|\omega)$ and use the identity

$$\nabla_\omega q_{-i}(\theta|\omega) = q_{-i}(\theta|\omega) \nabla_\omega \log q_{-i}(\theta|\omega)$$

in Equation (6.17) below

$$\nabla_\omega \log Z_i(\theta|\omega) = \frac{1}{Z_i(\theta|\omega)} \int\; t_i(\theta)\; q_{-i}(\theta|\omega)\; \nabla_\omega \log q_{-i}(\theta|\omega)\; d\theta \tag{6.17}$$

It is so apparent that Equation (6.17) is an expected value of $\nabla_\omega \log q_{-i}(\theta|\omega)$, that is,

$$E_{\hat{p}}\Big[\nabla_\omega \log q_{-i}(\theta|\omega)\Big]$$

Zeroing the gradient with respect to ω gives the conditions as follows;

$$\nabla_\omega \log Z_i(\theta|\omega) \approx \frac{1}{n} \sum_{k=1}^{n} \nabla_\omega \log q_{-i}(\theta_k|\omega) \tag{6.18}$$

where $\theta_k \sim \hat{p}$ which is the exact posterior distribution. We can therefore replace $\nabla_\omega \log Z_i(\theta|\omega)$ with the unbiased stochastic approximation of this gradient in Equation (6.18). We Denote this approximation by ζ_m. For example if $t_i(\theta)$ and $q_{-i}(\theta_{-i})$ are both of the distribution say Gaussian distributions then according to Minka (2001),

the first and second moments are as follows;

$$\nabla_{m_\theta} \log Z_i(\mathbf{m}_\theta, v_\theta) \approx \frac{1}{n}\sum_{k=1}^{n} \frac{\boldsymbol{\theta}_k - \mathbf{m}_\theta}{v_\theta} \tag{6.19}$$

then,

$$E_{\hat{p}}[\theta] \approx \mathbf{m}_\theta + v_\theta \nabla_m \log Z(\mathbf{m}_\theta, v_\theta) = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{\theta}_k \tag{6.20}$$

also,

$$\nabla_{v_\theta} \log Z_i(\mathbf{m}_\theta, v_\theta) \approx \frac{1}{n}\sum_{k=1}^{n} \frac{(\boldsymbol{\theta}_k - \mathbf{m}_\theta)^2 - v_\theta d}{2v_\theta^2} \tag{6.21}$$

We denote $\nabla_{v_\theta} \log Z_i(\mathbf{m}_\theta, v_\theta)$ as ζ_v in the following equation

$$E_{\hat{p}}[\theta^T\theta] - E_{\hat{p}}[\theta]^T E_{\hat{p}}[\theta] \approx v_\theta d - v_\theta^2(\nabla_m^T \nabla_m - 2\nabla_v \log Z(\mathbf{m}_\theta, v_\theta)) \tag{6.22}$$

Equation 6.22 can be written as follows;

$$E_{\hat{p}}[\theta^T\theta] - E_{\hat{p}}[\theta]^T E_{\hat{p}}[\theta] \approx v_\theta d - v_\theta^2(\zeta_m^T \zeta_m - 2\zeta_v) \tag{6.23}$$

6.4.2 EP-ADMM in low-dimensional space

In order to modify expectation propagation algorithm from deterministic viewpoint, we present a method based on alternating direction method of multipliers for the evidence of the posterior distribution which addresses the instability in EP. This method uses a ADMM of the gradient with respect to the approximation distribution q. The integration and gradient descent of the $\nabla_\omega \log Z_i(\theta|\omega)$ with ADMM for each observation become;

$$\begin{aligned} &\text{minimize} \quad \mathrm{KL}(\tilde{p}(\theta_{ij})||q(\theta_{ij})) \\ &\text{subject to} : m_{ij} = a; v_{ij} \geq b \end{aligned} \tag{6.24}$$

The Lagrangian formulation is

$$L(m_\theta, v_\theta, \alpha, \beta) = \mathrm{KL}(\tilde{p}(\theta_{ij})||q(\theta_{ij})) + \alpha^T(m_\theta - a) + \frac{\rho}{2}\|m_\theta - a\|_2^2 +$$

$$\beta^T(v_\theta - b) + \frac{\rho}{2}\|v_\theta - b\|_2^2 \tag{6.25}$$

Then, we compute

$$\nabla_{m_\theta, v_\theta, \alpha, \beta} L(m_\theta, v_\theta, \alpha, \beta), \tag{6.26}$$

where α, and β are the dual values of Equation (6.25);

$$Z_i(m_\theta, v_\theta) = \int t_i(\theta)\, q_{-i}(\theta|\omega) d\theta, \tag{6.27}$$

$$\int \frac{t_i(\theta)}{(2\pi v_i)^{d/2}} \exp\left\{\frac{1}{2v_\theta}(\theta - m_\theta)^T(\theta - m_\theta)\right\} d\theta, \tag{6.28}$$

Now, we find the gradient with respect to m_θ

$$\nabla_m \log Z_i(m_\theta, v_\theta) = \frac{1}{Z_i(m_\theta, v_\theta)} \nabla_m Z_i(m_\theta, v_\theta), \tag{6.29}$$

$$\nabla_m \log Z_i(m_\theta, v_\theta) = \int \frac{(\theta - m_\theta)}{v_\theta} \frac{t_i(\theta)}{(2\pi v_i)^{d/2}} \exp\left\{\frac{1}{2v_\theta}(\theta - m_\theta)^T(\theta - m_\theta)\right\} d\theta, \tag{6.30}$$

$$\nabla_m \log Z_i(m_\theta, v_\theta) = \frac{1}{v_\theta} \int \theta \frac{t_i(\theta)\, q_{-i}(\theta|\omega)}{Z_i(m_\theta, v_\theta)} d\theta - \frac{1}{v_\theta} \int m_\theta \frac{t_i(\theta)\, q_{-i}(\theta|\omega)}{Z_i(m_\theta, v_\theta)} d\theta, \tag{6.31}$$

$$\nabla_m \log Z_i(m_\theta, v_\theta) = \frac{1}{v_\theta} E[\theta] - \frac{m_\theta}{v_\theta}, \tag{6.32}$$

Make $E_{\tilde{p}}[\theta]$ the subject of formula, then equation 6.32 becomes

$$E_{\tilde{p}} = m_\theta + v_\theta \nabla_m \log Z_i(m_\theta, v_\theta), \tag{6.33}$$

then,

$$m_\theta^{new} = m_\theta + v_\theta \nabla_m \log Z_i(m_\theta^*, v_\theta) + \alpha + \rho(m_\theta - a), \tag{6.34}$$

We find the gradient with respect to v_θ from equation 6.26

$$\nabla_v \log Z_i(m_\theta, v_\theta) = \frac{1}{Z_i(m_\theta, v_\theta)} \int \frac{-d}{2} v_\theta^2 (\theta - m_\theta)^T(\theta - m_\theta) t(\theta)\, q_{-i}(\theta|\omega) d\theta, \tag{6.35}$$

$$-\frac{d}{2v_\theta} \frac{1}{Z_i(m_\theta, v_\theta)} \int t(\theta)\, q_{-i}(\theta|\omega) d\theta + \frac{1}{2v_\theta} \frac{1}{Z_i(m_\theta, v_\theta)} \int (\theta - m_\theta)^2 t(\theta)\, q_{-i}(\theta|\omega) d\theta, \tag{6.36}$$

$$-\frac{d}{2v_\theta}+\frac{1}{2v_\theta^2}E[\theta^2]-2\frac{m_\theta E[\theta]}{2v_\theta^2}+\frac{m_\theta^2}{2v_\theta}. \tag{6.37}$$

Using $m_\theta = E[\theta] - v_\theta \nabla_\theta log(Z_i(m_\theta, v_\theta))$

$$E[\theta^T\theta]-E[\theta]^T E[\theta] = v_\theta d - v_\theta^2\left[\nabla_m^T\nabla_m - 2\nabla_v \log(Z_i(m_\theta, v_\theta))\right] \tag{6.38}$$

then,

$$v_\theta^{new} = v_\theta d - v_\theta^2\left[\nabla_m^T\nabla_m - 2\nabla_v \log(Z_i(m_\theta^{new}, v_\theta))\right] + \beta + \rho(v_\theta - b) \tag{6.39}$$

$$\alpha^{new} := \alpha^k + \rho(m_\theta^{new} - a) \quad \text{and} \quad \beta^{new} := \beta^k + \rho(v_\theta^{new} - b) \tag{6.40}$$

6.4.3 Splitting EP algorithm for clutter problem

Algorithm 6 The General Splitting EP Algorithm

We present below the univariate version of the algorithm for SEP:

1. Initialize all of the approximating factors $\tilde{t}_i(\boldsymbol{\theta})$.
2. Compute the initial approximation $q(\theta)$ from the product of the approximating factors:
$$q(\theta) = \frac{\prod_i \tilde{t}_i}{\int \prod_i \tilde{t}_i d\theta},$$
3. Until all $\tilde{t}_i$ converge:
   (a) Choose a $\tilde{t}_i$ to refine
   (b) Remove $\tilde{t}_i$ from the approximation $q(\theta)$ by division:
$$q_{-i}(\theta) \propto \frac{q(\theta)}{\tilde{t}_i(\theta)}$$
   (c) Compute the tilted distribution $\tilde{p}_i(\theta)$ from q_{-i} and the exact factor t_i
$$\tilde{p}_i(\theta) = \frac{t_i(\theta)\, q_{-i}(\theta)}{Z_i},$$
   The normalizing factor is defined as
$$Z_i = \int t_i(\theta)\, q_{-i}(\theta) d\theta$$
   .
   (d) Since $\tilde{p}_i(\theta)$ is not in the chosen family $\mathcal{Q}$, we minimize the Kullback-Leibler divergence between the tilted distribution $\tilde{p}_i(\theta)$ and approximate distribution $q(\theta)$ that is
$$q^{new}(\theta) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}}\, \mathrm{KL}(\tilde{p}_i(\theta) || q(\theta)),$$
   Computing $\nabla_\omega \log Z_i(\theta)$ using Equation (6.10), and Equation (6.11) to update q by SSEP. Using Equation (6.34), Equation (6.39), and Equation (6.40), q is updated by EP-ADMM.
   (e) Compute the new approximate term:
$$\tilde{t}_i^{new}(\theta) = Z_i \frac{q^{new}(\theta)}{q_{-i}(\theta)},$$
4. Evaluate the approximation to the model evidence: $p(\mathcal{D}) \approxeq \int \prod_i \tilde{t}_i(\boldsymbol{\theta})\, d\boldsymbol{\theta}$

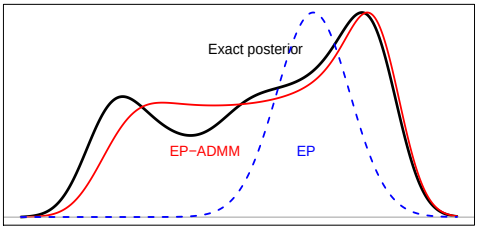


Figure 6.1: A complex posterior in the clutter problem produced by EP with ADMM and EP compared with the exact posterior.

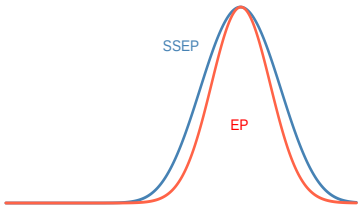


Figure 6.2: A complex posterior in the clutter problem produced by Stochastic search EP with Monte Carlo integration compared with EP.

6.4.4 Result of clutter problem

The strength of stochastic search expectation propagation over expectation propagation is the use of stochastic approximation within EP. The learning rate (lr) in Equation (6.12) determines how stable the stochastic search expectation propagation algorithm is, unlike expectation propagation which fails at the refining step. Learning rate controls the erratic nature of the expectation propagation algorithm, and this seems to be a viable improvement over EP. Moreover, it forces the variance not to reduce to negative value which often leads to EP's stability issue. EP was compared to four algorithms for approximate inference in Minka (2001) such as the Laplace's method, variational Bayes, Importance sampling where the prior is used as the importance sampling, and Gibbs sampling by introducing the hidden variables that determines if a data point is clutter or not. According to Minka (2001), EP competed well with other deterministic algorithms by approximating the posterior with a Gaussian. However, their performance improved substantially with more data i.e. the posterior is more Gaussian with more data. Figure (6.1) shows where the true posterior has three different modes (Black line). It can be seen that the EP-ADMM (Red line) captures at least two modes while EP captures one of the modes. This is not surprising as the EP has been termed as a unimodal algorithm. Figure (6.2) shows that the Stochastic search expectation propagation competes so well with EP.

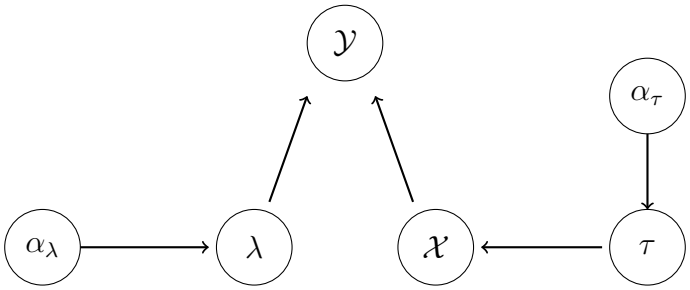


Figure 6.3: Graph networks connecting priors and their hyper-priors

Stochastic Search Expectation Propagation (SSEP) is a stochastic version of the EP algorithm and it can be seen that SSEP approximates EP at least properly well. We emphasize that EP-ADMM, and SSEP are all EP algorithms so the goal is not to find the most accurate algorithm but the generality and stability of the EP algorithm without a trade-off of the accuracy power of EP algorithm.

6.5 Model formation for Hierarchical Bayesian Model

6.5.1 Bayesian Formulation

We describe a Bayesian model for the data based on an inverse problem. Given data $\mathcal{D} = \{\mathcal{X}_n, \mathcal{Y}_n\}_{n=1}^N$, made up of $D-$dimensional vector $\mathcal{X}_n \in \mathcal{R}^D$ and corresponding target variables $\mathcal{Y}_n \in \mathcal{R}^D$. In the context of inverse problems, we only have access to a noisy version $\mathcal{Y}$ of the exact data $\mathcal{X}$. We assume that $\mathcal{Y}$ is obtained as $\mathcal{Y} = G\mathcal{X} + \xi$, where the $\xi \in \mathcal{R}^{\mathrm{N}\times\mathrm{D}}$ represents the noise in the data. Figure 6.3 shows graphical representation of the Bayesian hierarchical model which is mathematically represented as follows;

$$p(\mathcal{X}, \lambda, \tau|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{G}\mathcal{X}, \lambda)p(\mathcal{X}|\tau)p(\lambda)p(\tau)}{\int ... \int p(\mathcal{Y}|\mathcal{G}\mathcal{X}, \lambda)p(\mathcal{X}|\tau)p(\lambda)p(\tau)d\mathcal{X}d\lambda d\tau}, \tag{6.41}$$

Equation 6.41 has a large size of the integration which grows with the size of the reconstructed image $\mathcal{X}$. The standalone or independent variable is directly connected to both its mean $\mathcal{X}$ and variance λ. The priors are disconnected from each other but

directly connected to their respective hyper-priors. We consider a linear equation

$$\mathcal{F}(\mathcal{X}) = \mathcal{Y}, \tag{6.42}$$

where the map $\mathcal{F} : \mathcal{R}^{\mathrm{N}\times\mathrm{D}} \to \mathcal{R}^{\mathrm{N}\times\mathrm{D}}$, matrices $\mathcal{X} \in \mathcal{R}^{\mathrm{N}\times\mathrm{D}}$ and $\mathcal{Y} \in \mathcal{R}^{\mathrm{N}\times\mathrm{D}}$ refer to data formation mechanism, unknown parameter and the given data respectively.

6.5.2 EP via Monte Carlo integration called SSEP

In the context of high dimensional space, the proposed algorithm is applicable when the marginal likelihood involves mixture of family of distributions, for example Gaussian family of distributions and exponential distribution as we will encounter later. Splitting expectation propagation is an inference method that modifies expectation propagation method by looking into the intractable normalizing integral that makes the whole procedure unattainable at least for complex or hierarchical Bayesian model. Expectation propagation is vastly known for its fast and accurate properties but only in a less complex models. However, when the likelihood term and prior distribution are of different distributions from the exponential family, EP algorithm becomes unachievable. The splitting expectation propagation follows the EP setup but only differs by using different update technique. The update rule uses an idea proposed by John et al. (2011) for stochastic approach.

Let $\mathcal{Y}$ be an $N \times D$ matrix and $\mathcal{X}$ be an $N \times D$ matrix also. The likelihood for the model and the noise variance λ, with data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ is then

$$p(\mathcal{Y}|G\mathcal{X}, \lambda) = \prod_i \prod_j \mathcal{N}(\mathcal{Y}_{ij} \big| \mathcal{G}\mathcal{X}_{ij}, \lambda), \tag{6.43}$$

Also, we specify a Gaussian prior distribution for each entry $\mathcal{L}\mathcal{X}$ which is used as a matrix with zero mean and τ^2 as its variance. The Gaussian prior for $\mathcal{L}\mathcal{X}$ is as follows;

$$p(\mathcal{L}\mathcal{X}|\tau) = \prod_i \prod_j \mathcal{N}(\mathcal{L}\mathcal{X}_{ij} \big| 0, \tau), \tag{6.44}$$

The prior distributions for λ and τ are chosen to be an exponential distributions, i.e., $p(\tau) = \exp(\tau|\alpha_\tau)$ and also the noise variance to be $p(\lambda) = \exp(\lambda|\alpha_\lambda)$. The posterior

distribution for the parameters $\mathcal{X}, \lambda, \tau$ can then be obtained by

$$p(\mathcal{X}, \lambda, \tau|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{G}\mathcal{X}, \lambda)p(\mathcal{L}\mathcal{X}|\tau)p(\lambda)p(\tau)}{p(\mathcal{Y})}, \tag{6.45}$$

where λ, τ are unknown variances and $\mathcal{G}$ is a known matrix, $p(\mathcal{Y})$ is the normalizing factor and it will be denoted as Z throughout this work and in particular

$$Z = \int ....\int p(\mathcal{Y}|\mathcal{G}\mathcal{X}, \lambda)p(\mathcal{L}\mathcal{X}|\tau)p(\lambda)p(\tau)d\mathcal{X}d\lambda d\tau, \tag{6.46}$$

where $\mathcal{X}$ is $\mathcal{N} \times D$ dimension and $\mathcal{Y}$ is $\mathcal{N} \times D$ dimension. Next, we choose an approximating families. Here, the approximate distributions are different distributions of the exponential family. For instance,

$$q(\mathcal{X}, \lambda, \tau) = \left[\prod_{i=1}^{m}\prod_{j=1}^{n}\mathcal{N}(\mathcal{X}_{ij}\Big|mx_{i,j}, vx_{i,j})\right]\exp(\lambda|\alpha_\lambda)\exp(\tau|\alpha_\tau), \tag{6.47}$$

$\mathcal{X} \sim \mathcal{N}(mx_{i,j}, vx_{i,j})$, $\lambda \sim \exp(\alpha_\lambda)$ and $\tau \sim \exp(\alpha_\tau)$. The approximation parameters $mx_{i,j}, vx_{i,j}, \alpha_\tau$ and α_λ are determined by applying a stochastic search expectation propagation on the posterior in Equation (6.45). Finally, we sequence through and incorporate the terms t_i into the approximate posterior in Equation (6.47). At each step, we move from an old $q_{-i}(\mathcal{X}, \lambda, \tau)$ to a new $q(\mathcal{X}, \lambda, \tau)$.

The update rule is given in Equation (6.48) below;

$$\alpha_\tau^{new} \approx \alpha_{-ij} + v_{\alpha_{-ij}}\zeta_{\alpha_{-ij}}, \tag{6.48}$$

Where the mean $\frac{1}{K}\sum_{k=1}^{K}\tau^{(k)}$ is denoted as $\zeta_{\alpha_{-ij}}$ and variance as $v_{\alpha_{-ij}}$.
The update rule for λ is given in Equation (6.49) below;

$$\alpha_\lambda^{new} \approx \alpha_{-ij}^{\lambda} + v_{\alpha_{-ij}^{\lambda}}\frac{1}{K}\sum_{k=1}^{K}\lambda^{(k)}, \tag{6.49}$$

Taking the derivative of the normalizing factor with respect to ω concerning $\mathcal{X}$ as follows;

$$\nabla_{\omega} \log Z = \frac{1}{Z} \int_{\mathcal{X}_{i,j},\tau} \frac{t_{i,j}(L\mathcal{X}_{i,j},\tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j},\tau)} q_{i,j}(\mathcal{X}_{i,j},\tau) \nabla_{\omega} \log q_{i,j}(\mathcal{X}_{i,j},\tau) d\mathcal{X}_{i,j} d\tau, \tag{6.50}$$

At this juncture, we proceed with the use of Monte Carlo Integration

$$\frac{\delta_1}{\delta_0} = \sum_{n=1}^{N} \frac{t_{i,j}(L\mathcal{X}_{i,j},\tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j},\tau)} \nabla_{\omega} \log q_{i,j}(\mathcal{X}_{i,j},\tau) \Big/ \sum_{n=1}^{N} \frac{t_{i,j}(L\mathcal{X}_{i,j},\tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j},\tau)}, \tag{6.51}$$

when taking $\mathcal{X}$ into consideration then we fix τ and vice versa. So we update the parameter by taking the gradient step

$$\omega^{t+1} = \omega^{t} + lr * \frac{\delta_1}{\delta_0}, \tag{6.52}$$

One of the peculiarities of EP is the ability to minimize Kullback-Leibler divergence between the tilted posterior and the approximate posterior. The full derivatives are provided in the Appendix (F).

6.5.3 EP-ADMM in High dimensional space

EP becomes so vulnerable to numerical instability as discussed in the introduction. We introduce the Alternating direction method of multiplier (ADMM) algorithm to update the approximate posterior parameters.

$$\begin{aligned} &\text{minimize} \quad \mathrm{KL}(\tilde{p}(x_{ij}||q(x)_{ij})) \\ &\text{subject to} : m_{ij} \geq a; v_{ij} \geq b \end{aligned} \tag{6.53}$$

where a and b are constants. Then updating according to Minka (2001) is as follows;

$$m_x = \nabla_{m_{-ij}} \log Z_x + \alpha + \rho(m_{-ij} - a) \tag{6.54}$$

and

$$v_x = \nabla_{v_{-ij}} \log Z_x + \beta + \rho(v_{-ij} - b) \tag{6.55}$$

where

$$q(x) = \mathcal{N}(\mathcal{X}; m_x, v_x) \tag{6.56}$$

then the tilted distribution is

$$\tilde{p}(x_{ij}) = \frac{t_{ij}(x)q_{-ij}(x)}{\int_{\mathcal{X}_{ij}} t_{ij}(x)q_{-ij}(x)dx_{ij}} \tag{6.57}$$

Now the normalizing factor is

$$Z_x = \int_{x_{ij}} \mathcal{N}(\mathcal{Y}_{ij}; G\mathcal{X}_{ij}, \lambda)\mathcal{N}(\mathcal{X}_{ij}; m_{-ij}, v_{-ij})dx_{ij} \tag{6.58}$$

The update rule for the parameters m_x and v_x of $q(\mathcal{X})$ according to Minka (2001) with a method of multipliers are

$$m_x^{new} = m_{-ij} + v_{-ij}\frac{y_{ij} - gm_{-ij}}{v_{-ij}g^2 + \lambda}g + \alpha + \rho(m_i - a) \tag{6.59}$$

$$v_x^{new} = \frac{v_{-ij}\lambda}{v_{-ij}g^2 + \lambda} + \beta + \rho(v_i - b) \tag{6.60}$$

$$\alpha^{new} = \alpha^k + \rho(m_x^{new} - a) \quad \text{and} \quad \beta^{new} = \beta^k + \rho(v_x^{new} - b) \tag{6.61}$$

for full derivatives, Appendix (F) provides details.

6.5.4 EP-MCMC in High Dimensional Space

The EP-MCMC method provides an approach when the model is complex or the number of parameters is large and deterministic gradient-based techniques are infeasible and most importantly when the posteriors are not Gaussian. The Kullback-Leibler divergence between the tilted posterior and the approximated posterior can be addressed using the MCMC technique. In brief, the transitions in the Markov chain are designed so that an equilibrium distribution exists and is equal to the target distribution (tilted posterior). According to EP's strategy of moment matching, this is also possible but instead we update the moments of the approximated posterior with the moment of the tilted posterior from the Markov chain. We update the approximate term and compute the new cavity distribution from the new approximated posterior

and thereafter compute the new tilted posterior. With this cycle in mind, the advantage of EP-MCMC over the MCMC is that at each iteration in EP-MCMC the tilted posterior is updated with new cavity distribution thereby generating samples that are independent at each step of the algorithm. Unlike EP-MCMC, the major problem of MCMC is the dependence between samples generated at different iterations. This is due to a static target posterior. The EP setup for the parameters τ and λ is approached via Markov Chain Monte Carlo method.

$$\tilde{p}_{i,j}(\tau) \propto t_{i,j}(\tau) q_{-i,j}(\tau) \quad \text{and} \quad \tilde{p}_{i,j}(\lambda) \propto t_{i,j}(\lambda) q_{-i,j}(\lambda) \tag{6.62}$$

Algorithm 7 The General EP-MCMC Algorithm

Set an initial value for τ and λ

1. Initialize all of the approximating factors $\tilde{t}_i(\tau)$ and $\tilde{t}_i(\lambda)$.
2. Compute the initial approximation $q(\tau)$ and $q(\lambda)$ from the product of the approximating factors:
$$q(\tau) = \frac{\prod_i \tilde{t}_i}{\int \prod_i \tilde{t}_i \, d\tau} \quad \text{and} \quad q(\lambda) = \frac{\prod_i \tilde{t}_i}{\int \prod_i \tilde{t}_i \, d\lambda}$$
3. Until all $\tilde{t}_i$ converge:
   (a) Choose a $\tilde{t}_i$ to refine
   (b) Remove $\tilde{t}_i$ from the approximation $q(\tau)$ and $q(\lambda)$ by division:
   $$q_{-i}(\tau) \propto \frac{q(\tau)}{\tilde{t}_i(\tau)} \quad \text{and} \quad q_{-i}(\lambda) \propto \frac{q(\lambda)}{\tilde{t}_i(\lambda)}$$
   (c) Compute the tilted distribution $\tilde{p}_i$ from q_{-i} and the exact factor t_i
   $$\tilde{p}_i(\tau) \propto t_i(\tau)\, q_{-i}(\tau) \quad \text{and} \quad \tilde{p}_i(\lambda) \propto t_i(\lambda)\, q_{-i}(\lambda)$$
      i. Generate ϵ_τ and ϵ_λ from a Gaussian distribution
      ii. Generate a proposed new value $\tau^* = \tau + \epsilon_\tau$ and $\lambda^* = \lambda + \epsilon_\lambda$
      iii. Evaluate
      $$\alpha_\tau = \frac{\pi(\tau^*|\mathcal{X})}{\pi(\tau|\mathcal{X})} \quad \text{and} \quad \alpha_\lambda = \frac{\pi(\lambda^*|\mathcal{Y})}{\pi(\lambda|\mathcal{Y})}$$
      iv. Generate u_τ and u_λ from a uniform distribution $U(0,1)$
      v. If $\alpha_\tau > u_\tau$ and $\alpha_\lambda > u_\lambda$ then accept the proposals τ^* and λ^* and set $\tau = \tau^*$ and $\lambda = \lambda^*$
   (d) Update the approximated posterior from the inference of tilted posterior, this is similar to a moment matching in EP. $q^{new}(\tau)$ and $q^{new}(\lambda)$
   (e) Compute the new approximate term:
   $$\tilde{t}_i^{new}(\tau) \propto \frac{q^{new}(\tau)}{q_{-i}(\tau)} \quad \text{and} \quad \tilde{t}_i^{new}(\lambda) \propto \frac{q^{new}(\lambda)}{q_{-i}(\lambda)}$$
4. Evaluate the approximation to the model evidence:
$$p(\mathcal{X}) \approxeq \int \prod_i \tilde{t}_i(\tau) \, d\tau \quad \text{and} \quad p(\mathcal{Y}) \approxeq \int \prod_i \tilde{t}_i(\lambda) \, d\lambda \tag{6.63}$$

6.6 Implementation details

After incorporating the factors in Equation (6.57) for the first time, we constrained only mean and variance parameters of approximate posterior q of $\mathcal{X}$ for the purpose of this work using augmented direction method of multipliers. Because of the structure of the data presented in Figure (6.4) which is sparse, EP often breaks down when the algorithm produces variance parameter that is negative after incorporating one likelihood factor, especially at the update step of approximate term which involves division. A similar operation was reported in Minka (2001) when negative variances arise in Gaussian approximation factors. The inner loop of EP-ADMM takes about 7 minutes to run. We first present the results from the simulation study followed by the real data.

To effectively and successively apply EP to hierarchical models for the reconstruction of image from gamma camera, we introduced two strategies such as ADMM and stochastic search method. Stochastic search method is used where direct updates become impossible due to intractability of the normalizing factor i.e. often when the exact term and the cavity distribution are different, this was encountered with the updates of τ and λ. The hyper-priors for λ and τ are chosen to be an exponential distributions i.e., $p(\tau) = \exp(\tau|\alpha_\tau^0)$ and $(\lambda|\alpha_\lambda^0)$, with shape $\alpha_\lambda^0 = 0.001$ and $\alpha_\tau^0 = 0.001$. The approximation parameters m_x, v_x are determined by EP-ADMM while α_λ and α_τ are determined by applying MCMC method. We initialized $m_x = 0$ and $v_x = \tau$.

6.6.1 Monitoring Convergence of Splitting EP

To monitor convergence of the embedded Metropolis Hastings in EP, we run EP-ADMM $m = 10$ replications and $n = 1000$ to compute the Brook-Gelman statistic; Brooks & Gelman (1998) which compare the between-chain variance and within-chain variance. However, EP is generally known to be deterministic which is still preserved in EP-ADMM for the prior on $\mathcal{X}$ but the priors on λ and τ are randomly tuned. The reconstruction images presented are mean of the approximate posterior m_{ij}. To know if a convergence has been reached, Gelman & Rubin (1992a) suggested comparing m inferences computed from the m chains to the inferences computed by mixing together the mn draws from all the sequences.

To compute the between-chain and within-chain variances, let s_{jt} where $j = 1, ..., m$ and $t = 1, ..., n$ be the tth iterations of s in sequence j, the between-chain variance B/n and within-chain variance are given by

$$B/n = \frac{1}{m-1}\sum_{j=1}^{m}(\bar{s}_{j.} - \bar{s}_{..})^2 \tag{6.64}$$

and

$$W = \frac{1}{m(n-1)}\sum_{j=1}^{m}\sum_{t=1}^{n}(s_{jt} - \bar{s}_{j.})^2 \tag{6.65}$$

We also computed the *potential scale reduction factor* PSRF which is the variance of the pooled and within-chain inferences

$$R = \frac{\hat{V}}{\sigma^2}$$

R will be estimated by $\hat{R}$ because its denominator is unknown

$$\hat{R} = \frac{\hat{V}}{W} = \frac{m+1}{m}\frac{\hat{\sigma}_+^2}{W} - \frac{n-1}{mn} \tag{6.66}$$

where the pooled variance is given by

$$\hat{V} = \hat{\sigma}_+^2 + B/(mn) \tag{6.67}$$

and the estimated variance of the σ^2 given by the weighted average of B and W

$$\hat{\sigma}_+^2 = \frac{(n-1)}{n}W + \frac{B}{n} \tag{6.68}$$

Equation (6.68) is the unbiased estimate of the true variance σ^2. And $\hat{V}$ accounts for the sampling variability of the estimator which yields a pooled posterior variance estimate given in Equation (6.68). The PSRF is expected to be close to 1 for convergence to be guaranteed and each of the m sets of n simulated observations is close to the target distribution, however large value of $\hat{R}$ indicates no convergence or divergence and further simulations can be taken into consideration or the proposal distribution should be critically looked into.

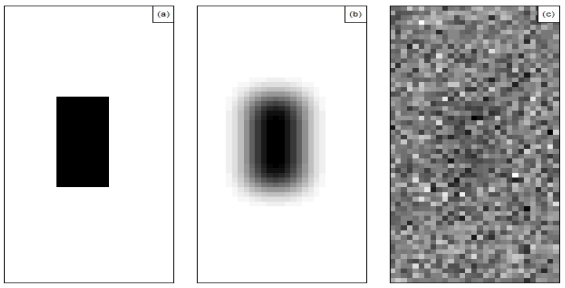


Figure 6.4: (a) True data: $\mathcal{X}$, (b) the mean: $\mathcal{G}\mathcal{X}$ and (c) Noisy data:$\mathcal{Y}$

6.7 An Application of SSEP on image

The synthetic data of a cylindrical image produced by a gamma camera is presented in Figure (6.4). The true image to be reconstructed is presented in Figure (6.4c) mixed with a noise. We evaluate SSEP in inverse problem with hierarchical Bayesian models, with data of cylindrical image from gamma camera. In SSEP, we retain the originality of the EP. However, we improve on the intractability of the normalizing factor by adopting the idea of John et al. (2011). The simulation was performed in [R (2019)]. The true parameters are the following; the precision is 100 and the true standard deviation is 0.1.

6.7.1 Reconstruction Results of SSEP and MCMC

We hereby present the reconstruction results of the SSEP and MCMC. Figure (6.5) shows the final output of SSEP after 1000 iterations while Figure (6.6) shows the outcome of the MCMC. The reconstruction result of SSEP outperforms that of MCMC in terms of sharpness of the image. Compared to the true image in Figure (6.4a), we observe that the posterior of $\mathcal{X}$ is well approximated by the SSEP method. In terms of the convergent time, i.e. time for both methods to reach convergence, SSEP is faster than MCMC. Unlike MCMC that needs about 500 chains as a burn-in period, SSEP does not require any thin-in or burn-in period. We observed a clear pattern in the residual plot shown in Figure (6.6b). This non-randomness may be due to the choice of the starting value of hyperparameters. The estimated precision for both SSEP and MCMC are provided in Figure (6.7) and Figure (6.6(c)) respectively. It

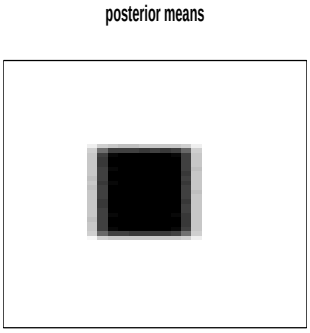


Figure 6.5: The reconstruction of the cylindrical image produced by SSEP

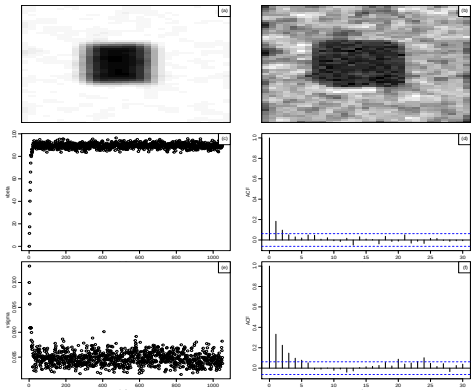


Figure 6.6: (a): The reconstruction image of the cylindrical image produced by MCMC.(b): relative error (c): The estimates of precision converge at about 85, (d): The estimates of standard deviation converges at 0.085.

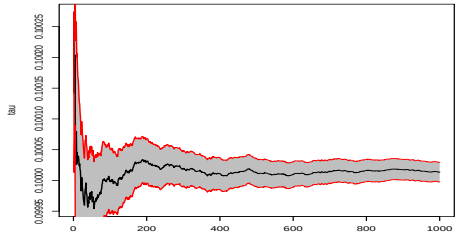


Figure 6.7: The precision is calculated from the plot as $\tau^{-2} = 100$. Since the variance is $\tau^2 = 0.01$

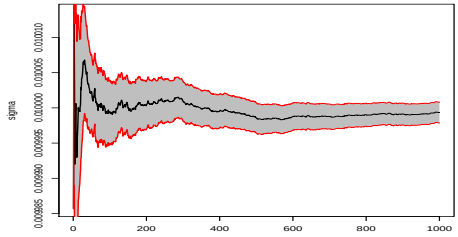


Figure 6.8: The variance output from the SSEP is 0.01 then the estimated standard deviation of SSEP is 0.1.

is not surprising to observe that SSEP produces the exact precision parameter value because of the high accuracy in the approximation of $\mathcal{X}$. Similarly, the estimate of variance parameter value are provided in Figure (6.8) and Figure (6.6(d)) for SSEP and MCMC respectively. Table (6.1) shows the comparison of the estimate of precision and variance parameter values for SSEP and MCMC with the true parameters. We observe that SSEP provides a closer estimate of both precision and variance to the true parameters.

Table 6.1: The Accuracy of the Standard deviation and precision produced by SSEP and MCMC for prior τ and λ compared to the True values provided

Parameter	True-Value	SSEP	MCMC
τ^{-2}	100	100	85
λ	0.1	0.1	0.085

6.8 An Application of Splitting EP on Animal Image Reconstruction

The full description of the study on the γ-eye system evaluated in a proof-of-concept animal study using normal Webster Swiss Albino mice with average weights of $25g$ can be found in Maria et al. (2016). The author conducted the study on three different clinical radio-pharmaceuticals which was radio-labeled with Tc-99m and injected the mice via their tail vein. The first mouse presented in Figure (6.9a) was injected with $100\mu 1/7.5$ MBq $[^{99m}Tc]$ MDP, which is a suitable agent for bone imaging and static images were obtained at $1h$ post-injection (pi), $2h$ pi, $3h$ pi, and $4h$ pi. The second mouse was injected by $100\mu 1/7.5$ MBq $[^{99m}Tc]$ DMSA, which is a tracer for imaging the anatomical structure and functional process of the kidneys. The author performed dynamic imaging for the first hour after injection and then static images for 10 min were acquired every $1h$ up to $24h$ pi. The image presented in Figure (6.9b) is a 1-h dynamic study of a mouse injected with $100\mu l$ of $[^{99m}Tc]$ DMSA radio-tracer. Then, static images were acquired at different time intervals. Figure $(6.9c)$ shows a third mouse injected with $100\mu 1/5.6$ MBq $[^{99mTc}]$ MIBI for heart perfusion. Dynamic imaging was performed for $2h$ pi and 10 min static images were obtained up to 5 h pi.

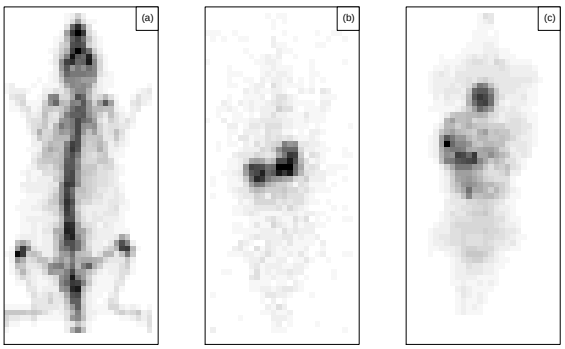


Figure 6.9: *(a)*: Mouse injected with $[^{99m}Tc]MDP$ at 4 h pi (15-min) acquisition time. (b): A mouse injected with ^{99m}Tc DMSA at $1h$ (10-min scan), 3 h (10-min scan), 5 h (10-min scan), 6h (10-min scan), and 24 h pi (30-min scan), (c): A static image of the mouse injected with the $[^{99m}Tc]$ MIBI at different intervals of time.

6.8.1 Parameter Setup for Splitting EP

The EP-ADMM algorithm was initialized with the following parameters; $\tau = \lambda = 0.01$ as a starting point for Metropolis Hasting algorithm embedded in EP to solve the intractability of the normalizing factor. We fixed the hyper-priors to $\tilde{\alpha}^{\tau}_{ij} = 2$, $\tilde{\alpha}^{\lambda}_{ij} = 1$ for the approximate term distribution and $\alpha^{\tau}_{ij} = \alpha^{\lambda}_{ij} = 10$ for the approximate posterior. According to the update of EP, we computed the cavity distribution parameters α^{τ}_{-ij} and α^{λ}_{-ij} just once to initialize the target distribution $p(\tau)$ and $p(\lambda)$. The outer loop of EP-ADMM consists of the EP-MCMC, i.e. tilted posteriors of τ and λ are hereby assumed to be intractable and thereby approximated using MCMC technique. We note that EP-MCMC has some advantages over ordinary MCMC in terms of speed to convergence, and accuracy. Also, we observed that due to different values of cavity distribution computed at different iterations of the algorithm, EP-MCMC does not need either burn-in period or thinning to achieve stationarity. To illustrate the accuracy of EP-ADMM for the image reconstruction and EP-MCMC for parameter recovery, we present also the results by a standard Metropolis Hasting MCMC with a sample of 1000 and burn-in period of 500. The random walk stepsize is chosen so that the acceptance rate is close to 0.234, which is widely known to be optimal; [Gelman et al. (1996)]. It is well known that the convergence of MCMC is very crucial.

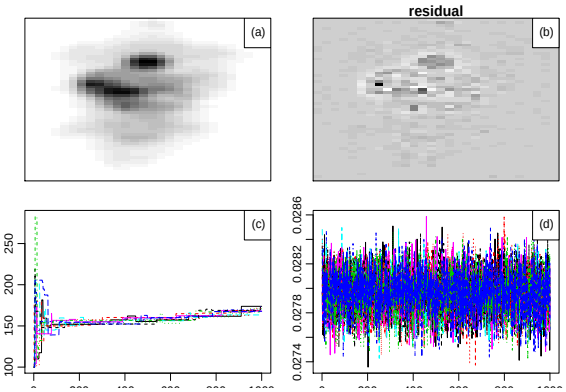


Figure 6.10: (a): 3h static_mibi_results from EP-ADMM, (b): relative error (c): estimates of τ converges at 158.48 from EP-MCMC, (d): estimates of λ converges at 0.028 from EP-MCMC.

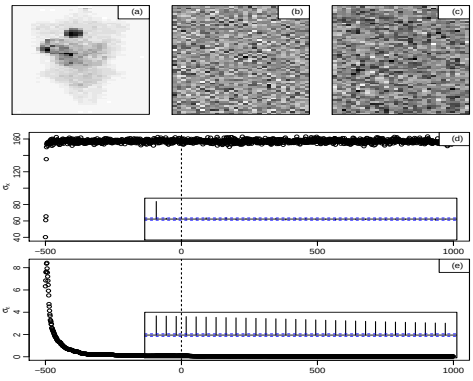


Figure 6.11: (a): 3h static_mibi_results from MCMC, (b)&(c): error & residual, (d): the estimate of σ_x converges at 159, (e): the estimates of σ_ϵ converges to about 0.9.

6.8.2 Reconstruction result for Mibi

First, we present the reconstruction result for the mouse injected with $100\mu 1/5.6$ MBq $[^{99mTc}]$ MIBI for heart perfusion. The estimate based on the Gaussian prior is shown in Figure (6.10). It can be seen that a well reconstruction is obtained with a substantive reduction in the noise. The result produced from Markov chain Monte Carlo is presented in Figure (6.11(a)). The reconstructions produced by both algorithms have a clear distinctions. First off, the reconstruction by EP-ADMM has a white background and smooth edges while the reconstruction produced by MCMC has a gray background and rough edges around the image reconstructed. This is a very important distinction because expectation propagation is well known for its fast computational speed. In contrast, MCMC is known for its slow convergence and it requires more computational time to reach a stationary distribution. Figure (6.10(b)) shows the error comparing the estimates with the data $\mathcal{Y}$. Comparing the MCMC results with the reconstruction produced in EP-MCMC, the parameter estimates of σ_x in MCMC is 160 while τ in EP-MCMC is about 158.48, similarly the parameter estimates of σ_ϵ is about 0.9 while that of EP-MCMC is 0.028.

The residual error shown in Figure (6.10(b)) comparing the estimates with the data, shows a foggy pattern. It is possible to produce similar random residual as shown in

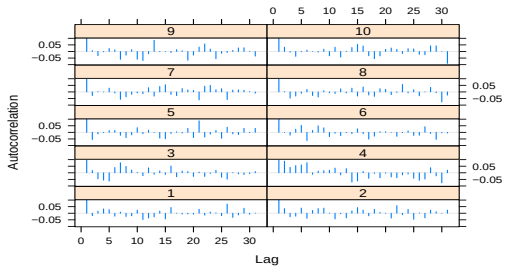


Figure 6.12: The Autocorrelation plot of the mouse injected with Mibi reagent for sigma; the EP-MCMC algorithm was run 10 time each of length 1000. The plot shows independence of chains at each replication.

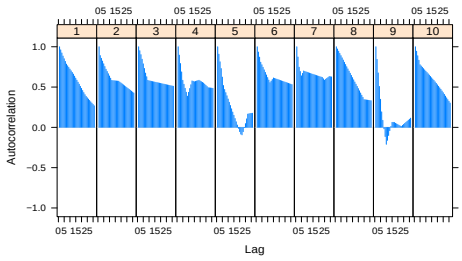


Figure 6.13: For the τ, the plot shows a strong dependence of chains at first four replications but at the fifth and ninth there is no correlation and the last replication shows a decline in correlation of the chains

Figure 6.11(b&c) by adjusting the value of α_λ and α_τ in the hyperparameter λ and τ respectively. This may be achieved by tweaking the values of the hyperparameters λ and τ. However, given our primary aim to produce a constant reconstruction, our focus is on the appropriate values of hyperparameters that produce smoother reconstruction.

Table 6.2: The mean estimates of the ten EP-MCMC chains for priors τ and λ of Mibi data

parameter	Posterior-Mean	$\hat{R}$	$\hat{V}$	W	B/n	σ_+^2
τ	158.48	1.06	173.58	98.53	68.32	166.75
λ	0.028	1.00	$2.98e^{-08}$	$2.49e^{-08}$	$2.48e^{-08}$	$2.73e^{-08}$

These parameter estimates are very significant in producing the clearer and smooth image. Given our primary aim to produce a clear and smoother image, which means larger τ in EP-MCMC and σ_x in MCMC would be preferred and lower values of σ_ϵ in MCMC and λ in EP-MCMC would be also be preferred. The autocorrelations of the estimates produced by MCMC are nested in their respective plot and it can be seen that the autocorrelations have been controlled by the well set-up of MCMC such as burn-in of 500 samples and thin-in of 1-of-10 samples. However, there seems to be a high correlation in Figure (6.11(e)) which might be reduced over time with large iteration. The computed potential scale reduction factors $\hat{R}$ for τ is 1.06, with upper C.I to be 1.12 and for λ is 1 with upper C.I to be 1 as shown in Table (6.2). Here, the potential scale reduction factor for λ is exactly 1 which indicates strong convergence

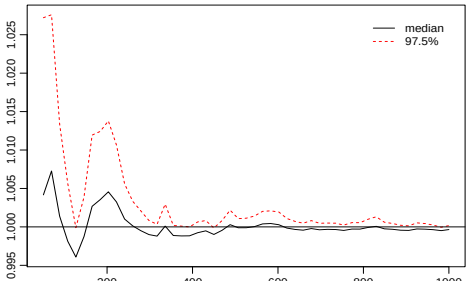


Figure 6.14: Iterative PSRF Plot for λ in Mibi image data (from $m = 10$ parallel sequence and $n = 1000$). The convergence starts at about 300 iterations till the end of the iteration.

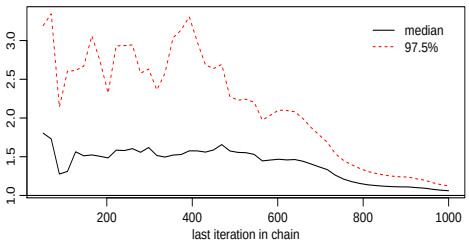


Figure 6.15: Iterative PSRF Plot for τ in Mibi image data (from $m = 10$ parallel sequence and $n = 1000$). The convergence starts at About 800 iteration till the end of the iteration.

across the 10 runs. According to Gelman & Rubin (1992a), large $\hat{R}$ can be interpreted as a need for more simulations to further reduce the estimate of the variance $\hat{\sigma}^2$ or to increase the W. The closeness to 1 indicates that each m sets of n simulated samples is close to the target distribution. However, potential scale reduction factor for τ is a bit more than one which is in tandem with further plots below. The simulation result agrees with the result from MCMC. Figure (6.12) shows the correlation between the chains of λ for each run. It can be seen that there is no autocorrection in the chains for each of the runs, this indicates that there is a convergence at every run which also contributes to the potential scale reduction factor of 1. Similarly, Figure (6.13) shows an autocorrelation plot of τ for the 10 runs. There is a high correlation among the chain of the first four runs, but at the fifth and ninth runs there is sharp reduction in the correlation while the tenth run shows a continuous decline in the correlation. Figure (6.16) and Figure (6.17) show the density of the marginal posterior distributions for both parameter λ and τ respectively. In each plot, the density summarizes the posterior sample. The thick horizontal bar at the bottom shows the posterior 95% credible interval and the maximum value is the posterior mean. The posterior estimate for λ is $\hat{\lambda} = 0.028$, with the credible interval of $(0.0275, 0.0285)$. The posterior estimate for τ is $\hat{\tau} = 158.48$ with the credible interval of $(150, 170)$.

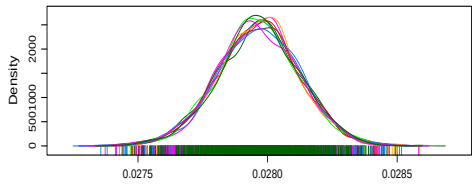


Figure 6.16: Marginal posterior distribution of the parameter λ for Mibi data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at 0.028 is the posterior mean.

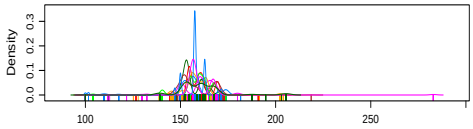


Figure 6.17: Marginal posterior distribution of the parameter τ for Mibi data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at about 160 is the posterior mean.

6.8.3 Reconstruction result for Mouse Data

Now, the reconstruction result for the mouse injected with $100\mu 1/7.5$ MBq $[^{99m}Tc]$ MDP, to X-ray the bone imaging is presented in Figure (6.18) and Figure (6.19) for splitting EP and MCMC respectively. Figure (6.18(a)) shows the estimate of the true image reconstructed from the noise by EP-ADMM. it can be seen that the image has a clear background with slightly rough edges on the image. The residual error shown in Figure (6.18(b)) comparing the estimates with the data, shows no randomness. Similar comment can be made about the residual shown in Figure 6.19(c&d). It is possible to reduce the patterns in the residual by adjusting the value of α_λ in the hyperparameter λ. This may lead to introducing another noise into the posterior mean estimate. Given that our primary aim is to produce a constant reconstruction, the focus is on the appropriate values of hyperparameter. Figure (6.18(c)) and Figure (6.18(d)) show the estimates of the parameter which converges to around 165.44 and 0.019 produced by EP-MCMC. Figure (6.19) on the other hand, shows a result produced by MCMC which is seen to have slightly similar results with EP-ADMM with a difference of blurry background. This might be as a result of long chain needed to reach convergence. Also, in Figure (6.19(e)), it can be seen that the samples are highly correlated even at 1500 iteration.

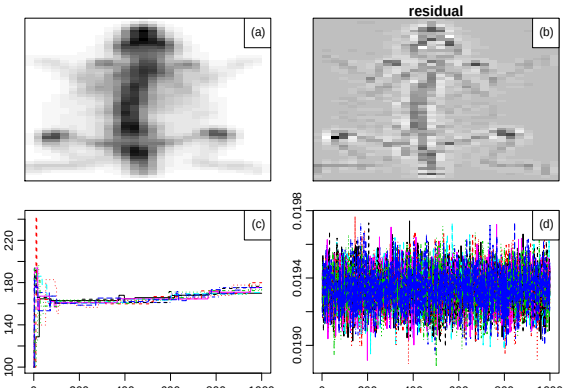


Figure 6.18: (a): mdp_1h_mouse_results from EP-ADMM, (b): relative error (c): estimates of τ which converges at 165.443., (d): estimates of λ converges at 0.019.

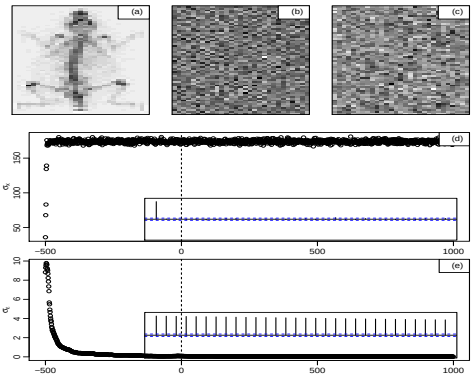


Figure 6.19: (a): mdp_1h_mouse_results from MCMC (b)&(c): error & residual, (d): the estimate σ_x converges at 160, (e): the estimates σ_ϵ converges to about 0.9.

To reduce the correlation, the number of n samples to thin might be increased that is, the number of samples to be discarded for each kept sample. Figure (6.18(c)) shows the trace plot for the parameter τ. It can be seen that the acceptance rate is quite low, this might be due to the choice of proposal distribution. Also, in Figure (6.19(e)) the σ_ϵ chains produced by MCMC have high correlation which doesn't show any sign of reduction. On the contrary, the observations σ_x of MCMC exhibit independence in the chains. However, all the parameters from EP-MCMC and MCMC converge to a very similar value with slight differences. We further establish the convergence of the τ and λ for EP-MCMC.

Table 6.3: The mean estimate of the ten EP-MCMC chains for prior τ and λ of mouse data

parameter	Posterior-Mean	$\hat{R}$	$\hat{V}$	W	B/n	σ_+^2
τ	165.44	1.02	56.84	56.58	0.00031	56.83
λ	0.019	1.00	$1.21e^{-08}$	$1.21e^{-08}$	$1.14e^{-11}$	$1.21e^{-08}$

Table (6.3) shows some statistical inference to establish the convergence properties of the EP-MCMC parameters. The potential scale reduction faction of τ is 1.02 which is still within the acceptable range of 1.

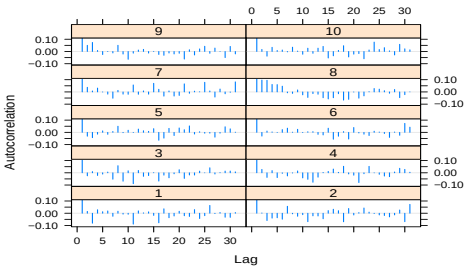


Figure 6.20: The Autocorrelation plot of the mouse injected with Mdp reagent; the EP-MCMC algorithm was run 10 time each of length 1000. The plot shows independence of chains at each replication.

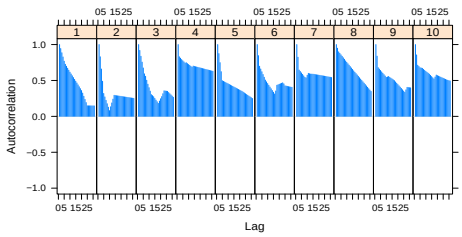


Figure 6.21: The plot shows a reduction in dependence of chains at first three replications but at the fifth and ninth there is no correlation and the last replication shows a decline in correlation.

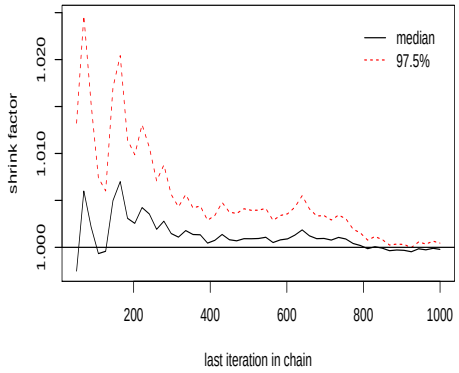


Figure 6.22: Iterative PSRF Plot for λ in Mouse image data (from $m = 10$ parallel sequence and $n = 1000$). About 800 iterations, the convergence starts till the end of the iteration.

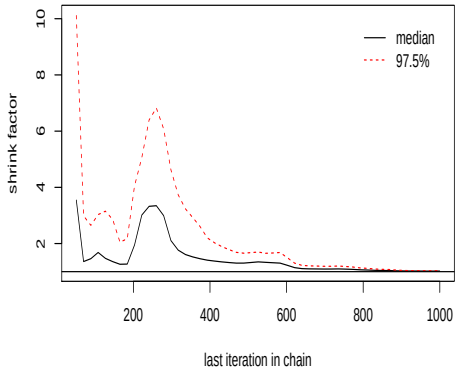


Figure 6.23: Iterative PSRF Plot for τ in Mouse image data (from $m = 10$ parallel sequence and $n = 1000$). About 600 iterations, the convergence starts till the end of the iteration

This can be interpreted as the chain represents the target distribution of the parameter τ. Also, the PSRF for λ is exactly 1.

The within-chain, between-chain variances, and pooled variance for τ are $W = 56.58$, $B/n = 3.1 \times 10^{-4}$, and $\hat{V} = 56.84$ respectively. Similarly, the within-chain, between-chain variances, and pooled variance $W = 1.21 \times 10^{-8}$, $B/n = 1.14 \times 10^{-11}$, and

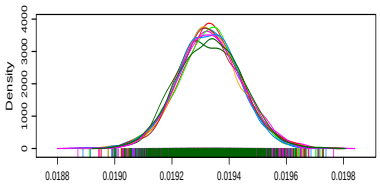


Figure 6.24: Marginal posterior distribution of the parameter λ for Mouse data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at 0.019 is the posterior mean.

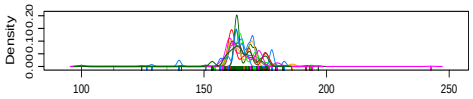


Figure 6.25: Marginal posterior distribution of the parameter τ for Mouse data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at about 166 is the posterior mean.

$\hat{V} = 1.21 \times 10^{-8}$ respectively. It can be seen that according to the interpretation by Gelman & Rubin (1992a), the equality of pooled variance $\hat{V}$ and W for both parameters indicates the existence of convergence. Figure (6.22) gives a visual interpretation of the values presented above in Table (6.3). The shrinking factor for λ in Figure (6.23) was above 1 at the initial stage of the iteration, but on getting to around 800 iterations the equality between the between-chain and within-chain becomes so evident. This indicates that 1000 iterations is sufficient to provide convergence for parameter λ. Likewise, the shrinking factor for the τ at the first 100 iterations is 10 which means the chains still need a lot of iterations to bring reduction in the shrinking factor. At about 600 iterations, the shrinking factor is on the acceptable line of 1 which according to the Table (6.3) is 1.02.

Figure 6.24 and Figure 6.25 show the density of the marginal posterior distributions for both parameter λ and τ respectively. In each plot, the density summarizes the posterior sample. The thick horizontal bar at the bottom shows the posterior 95% credible interval and the maximum value is the posterior mean. The posterior estimate for λ is $\hat{\lambda} = 0.019$, with the credible interval of $(0.01885, 0.0198)$. The posterior estimate for tau is $\hat{\tau} = 165.44$ with the credible interval of $(100, 250)$.

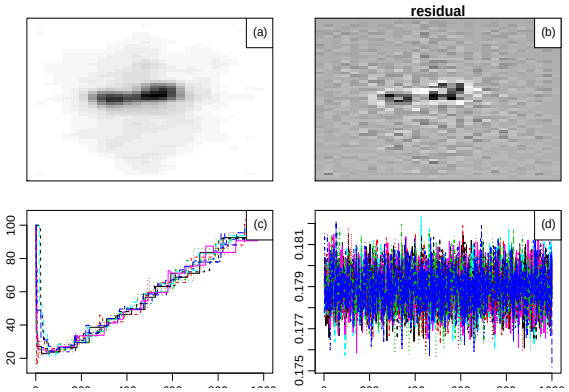


Figure 6.26: (a): dmsa_1h_mouse_results from SEP, (b): relative error (c): estimate τ is 58.9, (d): estimate λ is 0.179

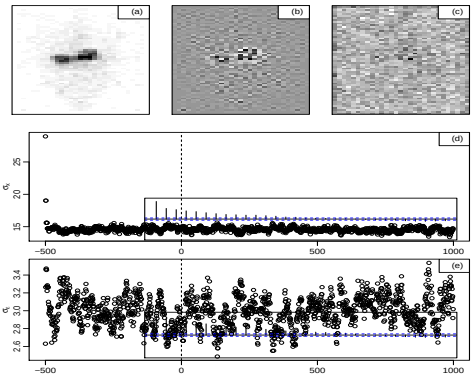


Figure 6.27: (a): MCMC Dmsa (b)&(c): error & residual, (d)&(e): the estimate of σ_x & σ_ϵ converges at 10 and 3.0

6.8.4 Reconstruction result for DMSA Data

Now, moving to the reconstruction of the mouse injected with DMSA. The results produced by splitting EP and MCMC are quite similar in Figure (6.26(a)), Figure (6.26(b)), Figure (6.27(a)), and Figure (6.27(b)) respectively. Given that we require clear and smooth image reconstruction, EP-ADMM produces a smooth image reconstruction with a clear and white background. In contrast, MCMC produces a white and clear background with sharp surface coupled with slightly rough edges. The residual error shown in Figure (6.26(b)) and Figure (6.27b) comparing the estimates with the data, show similar pattern. Adjusting the value of α_λ and α_τ in the hyperparameters λ and τ respectively may produce random errors.

In Figure (6.26(d)), the mean estimate of the posterior for λ across the 10 replication is 0.179, which can be seen in the trace-plot. Figure (6.26(c)) does not show any

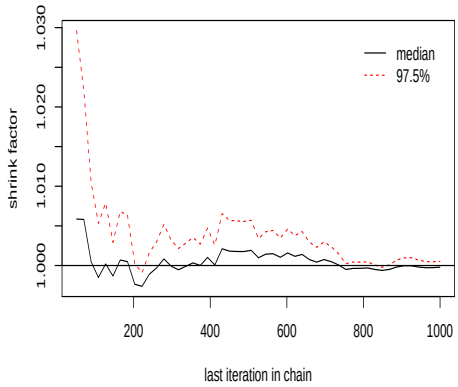


Figure 6.28: Iterative PSRF Plot for λ in Mouse image data (from $m = 10$ parallel sequence and $n = 1000$). After about 800 iteration convergence starts till the end of the iteration.

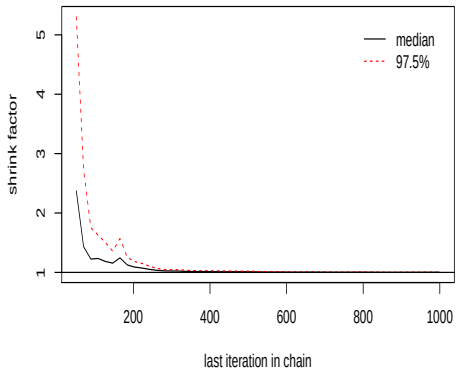


Figure 6.29: Iterative PSRF Plot for τ in Mouse image data (from $m = 10$ parallel sequence and $n = 1000$). About 200 iteration convergence starts till the end of the iteration.

pattern of convergence even after 1000 iterations. This however may be due to the choice of proposal distribution.

Table 6.4: The mean estimate of the ten EP-MCMC chains for prior τ and λ of DMSA data

parameter	Posterior-Mean	$\hat{R}$	$\hat{V}$	W	B/n	σ^2_+
τ	58.82	1.00	59.89	59.87	0.75	59.89
λ	0.179	1.00	$7.60e^{-07}$	$7.60e^{-07}$	$8.93e^{-10}$	$7.60e^{-07}$

Figure 6.30 and Figure 6.31 show the density of the marginal posterior distributions for both parameter λ and τ respectively. In each plot, the density summarizes the posterior sample. The thick horizontal bar at the bottom shows the posterior 95% credible interval and the maximum value is the posterior mean. The posterior estimate for λ is $\hat{\lambda} = 0.179$, with the credible interval of $(0.176, 0.182)$. The posterior estimate for tau is $\hat{\tau} = 58.82$ with the credible interval of $(1, 102)$.

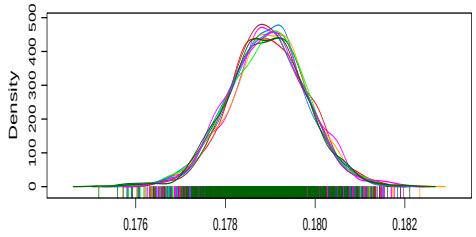


Figure 6.30: Marginal posterior distribution of the parameter λ for DMSA data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at 0.179 is the posterior mean.

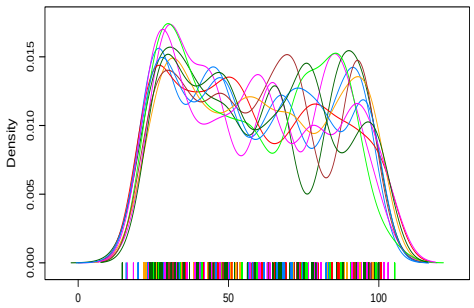


Figure 6.31: Marginal posterior distribution of the parameter τ for DMSA data. The tick horizontal bar signifies the posterior 95% credible interval and the maximum value at about 59 is the posterior mean.

6.8.5 Reconstruction result of four Circles Data

Finally, Figure (6.33(a)) presents the reconstruction result for the image of four circles. It can be seen that EP-ADMM produces a clear and sharp reconstruction image, while MCMC produces a scaly and gray background as shown in Figure (6.34(a)). Comparing the parameter estimates of MCMC with EP-MCMC, the parameter estimates of σ_x in MCMC is 160 while τ in EP-MCMC is about 200, similarly the parameter estimates of σ_ϵ is about 0.5 while that of EP-MCMC is 0.023. The residual error shown in Figure (6.33(b)) comparing the estimates with the data, shows a very clear pattern. Random residual as shown in Figure 6.11(b&c) can be produced by using an appropriate values of the hyperparameters λ and τ. However, given our primary aim to produce a constant reconstruction, this remains a little concern and it can be further checked in the future. The computed potential scale reduction factors $\hat{R}$ for τ is 1.1, upper C.I is 1.12, and λ is 1 with upper C.I as 1 are shown in Table (6.5).

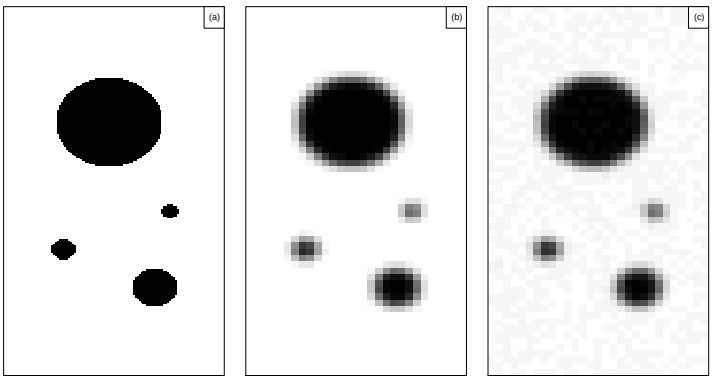


Figure 6.32: (a) Original image of circles to be reconstructed: $\mathcal{X}$, (b) the mean: $\mathcal{G}\mathcal{X}$ and (c) Noisy data:$\mathcal{Y}$

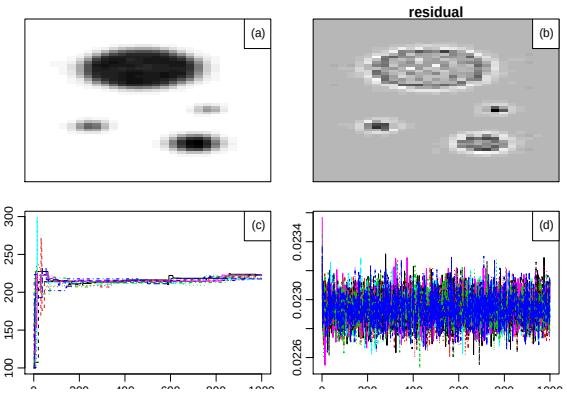


Figure 6.33: (a): The reconstruction image of four circles by EP-ADMM, (b): relative error (c): estimates τ converges at 200 from EP-MCMC, (d): estimates λ converges at 0.023 from EP-MCMC

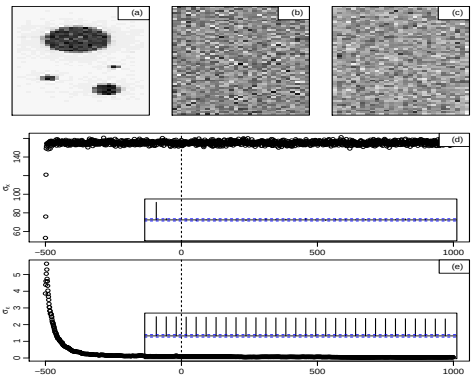


Figure 6.34: (a): The reconstruction image of four circles produced by MCMC (b)&(c): error & residual, (d): the estimate of σ_x converges at 150, (e): the estimates of σ_ϵ converges to about 0.5

Here, the potential scale reduction factor for λ is exactly 1 which indicates strong convergence across the 10 runs. Figure (6.35) shows the correlation between the chains of λ for each run. It can be seen that there is no autocorrection in the chains for each of the runs, this indicates that there is a convergence at every run which also contributes to potential scale reduction factor of 1. Similarly, Figure (6.36) shows an autocorrelation plot of τ for the 10 runs at the initial stage of the runs. There is a high correlation among the chain of the first four runs, but at the fifth and ninth runs

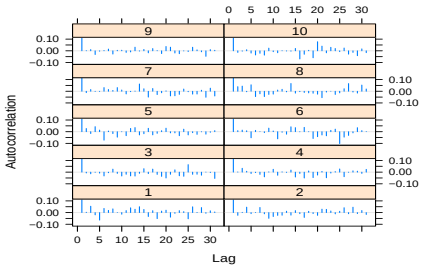


Figure 6.35: The Autocorrelation plot of the image of four circles; the EP-MCMC algorithm was run 10 time each of length 1000. The plot shows independence of chains at each replication.

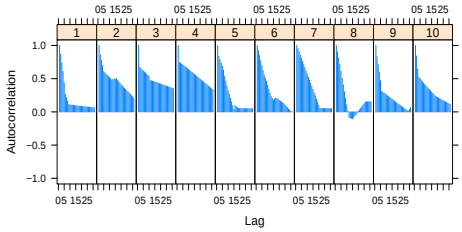


Figure 6.36: The plot shows a strong dependence of chains at first four replications but at the fifth and ninth there is no correlation and the last replication shows a decline in correlation

Table 6.5: The mean estimate of the ten EP-MCMC chains for prior τ and λ of Circles

parameter	Posterior-Mean	$\hat{R}$	$\hat{V}$	W	B/n	σ_+^2
τ	200	1.10	112.06	110	2.16	112.06
λ	0.023	1.00	$9.84e^{-09}$	$9e^{-09}$	$1.41e^{-11}$	$9.84e^{-09}$

there is sharp reduction in the correlation while the tenth run shows a continuous decline in the correlation.

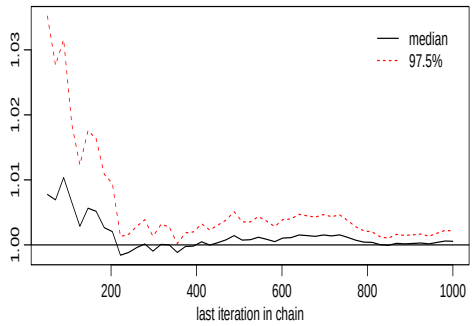


Figure 6.37: Iterative PSRF Plot for λ in image of four circle (from $m = 10$ parallel sequence and $n = 1000$). After about 200 iteration convergence starts with a bump at the middle but got stabilized around 300, till the end.

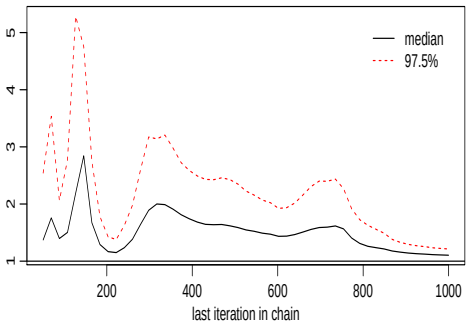


Figure 6.38: Iterative PSRF Plot for τ in image of four circles (from $m = 10$ parallel sequence and $n = 1000$). About 200 iteration convergence started and diverged in the middle but later converged at about 800 till the end of the iteration.

6.9 Summary

In this chapter we have presented an algorithm called Splitting Expectation Propagation for approximating hierarchical Bayesian models. Our algorithm focused on the problems of instability and intractability in EP especially at the refinement stage where it often fails and its inability to handle hierarchical Bayesian models due to problem of rigorous moment matching especially when the prior and likelihood family distributions differ. The modification was employed from stochastic and deterministic standpoints.

From the stochastic point, we adopted the technique by John et al. (2011) which has led us to use Monte Carlo integration for the intractable integration of the EP. This stochastic approach to EP is what we called Stochastic Search Expectation Propagation (SSEP). SSEP was used for the image reconstruction and it competed well with the MCMC in that SSEP produced well approximated parameter estimates. The second approach from the deterministic viewpoint is the introduction of both ADMM and MCMC to EP. This is what we referred to as EP-ADMM and EP-

MCMC respectively. EP-ADMM was used to handle the image reconstruction and EP-MCMC was used for estimating the hyperparameters. According to the results presented EP-ADMM produced very clear, sharp and white background. While in the same algorithm, EP-MCMC is the use of MCMC at the refinement stage of EP. EP-MCMC achieved low variances when compared to the ordinary MCMC. Splitting EP in these viewpoints is the umbrella over both SSEP, EP-ADMM, and EP-MCMC. Splitting EP was able to solve the problems of intractability and inflexibility of EP to hierarchical Bayesian models. Splitting EP can therefore be seen as an alternative EP for image analysis.

The main limitation of SSEP is that it required large number of samples to fulfill the central limit theorem due to the Monte Carlo integration. As a result of the large samples, SSEP is slow to converge. Moreover, the limitation of the EP-MCMC differs from the ordinary MCMC in that the inner loop EP-ADMM that contributes to the EP-MCMC for the hyperparameter is efficiently fast. On the contrary, ordinary MCMC for the prior $\mathcal{X}$ will be slow resulting in slow convergence in total. As a future direction, we look forward to handling the hyperparameters with Variational Inference. This method would make perfect sense as the approximate posterior from the prior $\mathcal{X}$ will be used as the auxiliary posterior distribution factorized over both the hyperparameters λ and τ which can be seen as a Coordinate Ascent Variational Inference (CAVI) [Blei et al. (2018)].

Chapter 7

Conclusion

7.1 Discussion

In this thesis we have worked in the Bayesian learning context, inference, clustering, classification, and image recognition using the variants of variational approximations. We have described a general framework of variational Bayesian learning and show how it can be applied to several models of interest in high-dimensional data. We have approached variational Bayesian learning from both deterministic and stochastic viewpoint. The family of variational approximations used in this thesis are Expectation-Maximization, Variational Bayes, and Expectation propagation algorithms. Then we discussed their relationship.

7.2 Summary of contributions

The aim of this thesis has been to investigate the variational Bayesian methods for approximating Bayesian inference and learning in a variety of statistical models used in machine learning applications. We have used the term variational Bayesian as a generic name for any optimization method used in this work. For example, The family of variational Bayesian used in this thesis are Expectation-Maximization, Variational Bayes, and Expectation propagation. We have approached Expectation-Maximization and Expectation propagation from both deterministic and stochastic standpoints. Chapter 1 provided a general background for machine learning and re-

viewed some machine learning terms. We have showed that in situations where the parameters of the model are unknown the correct Bayesian procedure is to integrate over this uncertainty to form the marginal likelihood of the model. We explained that the marginal likelihood is intractable to compute for almost all interesting models. We discussed the supervised, semi-supervised, and unsupervised. Chapter 1 also reviewed some basics of probabilistic inference in the Bayesian context. We provided a general introduction to image reconstruction and how it relates to inverse problems which is tackled by Bayesian approach.

We reviewed the exact Bayesian inference which is extremely intractable in high-dimensional space. We also reviewed a number of current methods for approximating the marginal likelihood, Markov Chain Monte Carlo (MCMC), Monte Carlo method, and Importance sampling. We note that Maximum a posterior estimate may not be a representative of the posterior mass at all. Moreover we noted that the MAP optimization does not produce the same prediction using the MAP estimates with the same model and priors, but with different parameterizations. This means that MAP is basis dependent. We also discussed a variety of sampling methods, and noted that these are guaranteed to produce an exact answer for the marginal likelihood only with an infinite number of samples, and impractically long sampling runs to obtain accurate and reliable estimates.

In chapter 2, we presented the family of variational Bayesian for approximating the marginal likelihood of the exact Bayesian model. We first treated the standard expectation-maximization (EM) algorithm for learning parameters as a member of the family of Variational Bayesian method. EM can be interpreted as a variational optimization of a lower bound on the likelihood of the data. We note the similarity between EM and VB through the minimization of the Kullback-Leibler divergence but dissimilar only through a deterministic parameters. In this optimization, the E step can be restricted to a particular family of distributions in which case the bound is loose. The amount by which the bound is loose is exactly the Kullback-Leibler divergence between the latent variable posterior and the exact posterior. We then provided the general background of the variational Bayes (VB). We note that VB is the generalized EM by treating the hidden parameters as a random variable which leads to a hierarchical Bayesian model. We reviewed the mean-field variational family

which is a factorized form of the VB. We discussed the appropriate algorithm and when a statistician should choose between VB and MCMC. Finally, we gave a general review of expectation propagation algorithm. We note the different between VB and EP. We also treated EP as a member of Variational Bayesian by minimizing the inclusive/reverse of Kullback-Leibler divergence.

In chapter 3, we investigated the application of cluster weighted models (CWMs) in high-dimensional space. First, we gave a general background study of CWM. We discussed how CWM transitioned from finite mixture model (FMM) due to a problem of assignment independence. We noted the limitations of CWM in high-dimensional space. Moreover, we noted that eigenvalue decomposition is not enough for CWM to handle high-dimensional data. We therefore discussed a powerful dimensionality reduction technique called t-distributed stochastic neighbor embedding (tSNE). tSNE is a nonlinear transformation of high-dimensional data to a low-dimensional representation which can preserve the hidden structure of the data. We noted also that the clustering or classification power of CWM is hampered by the dimensionality of the data. We integrated the tSNE with CWM in order to increase the classification power of CWM in the high-dimensional space. We applied CWM-tSNE on both moderate-dimensional data and high-dimensional data. In particular we analyzed and clustered the Epileptic seizure recognition whose goal was to discover the hidden structure by the information criteria. The plausible future research is to propose a multivariate CWM where the response variables are multivariate Gaussian distribution.

In chapter 4, we proposed a new member of CWM appropriate for multiclass response variable called Multinomial CWM (MCWM). MCWM combat the inability of CWMs to handle categorical data and failure in the presence of high-dimensional data. The proposed model allowed for the nonlinear dependency in the mixture components using the multinomial logit regression or softmax regression. In addition MCWM considered multinomial distribution for the conditional distribution of the response variable given covariates. We derived the identifiability conditions for MCWM. First, we approached the method of parameter estimation using EM algorithm from two viewpoints. We developed EM-IRLS for estimating the parameters of MCWM. At the E step, we updated the complete log-likelihood while at the M step, we adopted the Newton-Raphson algorithm to maximize the parameters. However,

the limitation associated with EM-IRLS hindered the scalability of the algorithm to high-dimensional space. To solve this problem, we developed the EM-SGD. EM-SGD algorithm employed the Stochastic Gradient Descent (SGD) algorithm at the M step to estimate the parameters. We analyzed both simulated and real data and compared the results of the proposed model to other models. We evaluated the classification power of MCMW using both confusion matrix and ARI and we compared to other existing models such as Logistic regression, NTgrowth, C4, Classit models. According to the result provided, We have concluded that MCWM was superior to other existing models. Therefore MCWM can further be given a consideration for classifying multi-class data. With the help of EM-SGD, we scaled MCWM to high-dimensional space and classified handwriting image data. This is novel in the field of CWMs and the result was comparably accurate. To achieve high accuracy in machine learning models, the hyperparameters must be tuned using different types of cross-validation methods. Feasible future direction will be to investigate different type of cross-validation method on MCWM. Moreover, a regularization technique should be employed.

The limitation of MCWM stemed from the method of parameter estimation. For example, EM-IRLS has a problem of matrix inversion. This has been tackled by a stochastic gradient descent algorithm. Also, MCWM does not take into account the problem of class imbalance in the data. Class imbalance is inevitable mostly in the medical field possibly due to a loss of information or censored information. Therefore, MCWM will tend to favor the class with large distribution and give a misleading result. We looked more into this problem in the next chapter.

In chapter 5, we tackled the problem of class imbalance using zero-inflation models. We proposed a model that addressed the presence of excess zeros in the data which often cause erroneous result leading to wrong decision making. We developed a model in the context of CWM and extended the Poisson CWMs (PCWM) by Ingrassia et al. (2015) to account for zero-inflation in the data. The model is called zero-inflated Poisson CWM (ZIPCWM). ZIPCWM has many models as special cases such as Poisson CWM [Ingrassia et al. (2015)], Generalized Zero-inflated Poisson regression mixture model [Hwa et al. (2014)], Zero-inflated Poisson distribution [Lambert (1992)], and Standard Poisson mixture model. We developed EM-IRLS due to the Poisson count for the response variable. We also derived the identifiability conditions for parameter

discovery. We further investigated the classification power on a count data whose intrinsic structure or class was unknown. First, we discovered the hidden cluster of the data. Afterward, we classified the observations according to the cluster discovered. We compared the classification power of ZIPCWM to both PCWM and FZIP, we discovered that FZIP performed the worst among the competing models. PCWM performed averagely but hampered by the problem of label switching. ZIPCWM performed the best as it took account of the extra zeros in the data and the covariates distribution had an advantage over FZIP.

In chapter 6, we focused on different type of variational Bayesian approximation called Expectation propagation (EP) algorithm [Minka (2001)]. EP instead minimizes the Kullback-Leibler divergence between the tilted distribution and the approximate distribution. We investigated EP in high-dimensional context and noted that it is infeasible due to its memory inefficiency. We also noted that EP proved to be difficult when both tilted distribution and approximate distribution are of different exponential family other than Gaussian distribution. Working with different distributions other than Gaussian, the Kullback-Leibler to be minimized can either be addressed by Gaussian quadrature which is an approximation of the definite integral of a function (KL divergence). We noted that this is however infeasible with multidimensional finite integrals. We therefore approached EP from the stochastic perspective called Stochastic search EP (SSEP). SSEP employed the idea of John et al. (2011). We employed the Monte Carlo approximation and SGD to minimize the KL divergence between the non-Gaussian tilted distribution and approximated distribution. The result was comparably accurate with the result produced by MCMC. However, we noted that there was a great impact of high-dimensional space on SSEP. This is due to the large number of samples needed to achieve convergence. To combat this problem, we proposed an unification of stochastic and deterministic of EP called Splitting EP algorithm (SEP). SEP algorithm used an alternating direction method of multiplier as a deterministic part which we called the EP-ADMM. EP-ADMM handled the image reconstruction in the hierarchical Bayesian model. The stochastic version used MCMC in EP called EP-MCMC which handled the parameters estimation. The images reconstructed were comparably sharper than the ones reconstructed by MCMC.

Appendix A presents an ongoing work on the hybridization of VB and EP. We discussed the hybridization of VB and EP with probabilistic Backpropagation for Bayesian Neural Networks. Although the work is ongoing but the mathematical derivations are presented in the Appendix A. We show here with an example that EP can be approached from VB. The work generalizes the work done by Jose & Ryan (2015) in a linear regression context. However, the new work is the binary classification using Binary Logistic regression in VBEP.

To conclude, I hope that this thesis has provided an accessible and coherent account of the widely applicable variational Bayesian approximation. We have derived a families of Variational approximation for varieties of statistical models. We have addressed two solid problems of CWMs in this thesis and the problem arising from the original EP for image reconstruction. The hope is that the experimental findings and insights documented in these chapters will stimulate and guide future research on variational approximation.

Appendix A

VBEP Algorithms with Probabilistic Backpropagation for Bayesian Neural Networks

A.1 Introduction

We propose a novel approach for nonlinear Logistic regression using a two-layer neural network (NN) model structure with hierarchical priors on the network weights. We present three variants of expectation propagation such as expectation propagation expectation maximum (EP-EM) approach, Variational Bayes-Expectation Maximization approach (VBEM), and Variational EP-EM approach for approximate integration over the posterior distribution of the weights, the hierarchical scale parameters of the priors and zeta. Using a factorized posterior approximation we derive a computationally efficient algorithm, whose complexity scales similarly to an ensemble of independent sparse logistic models. The approach can be extended beyond standard activation functions and NN model structures to form flexible nonlinear binary predictors from multiple sparse linear models. The effects of the hierarchical priors and the predictive performance of the algorithm are assessed using both simulated and real-world data. We consider a hierarchical Bayesian model with logistic regression likelihood and a Gaussian prior distribution over the parameters called weights and

hyperparameters. We work in the perspective of E step and M step for computing the approximating posterior and updating the parameters using the computed posterior respectively.

A.1.1 Main contribution

The goal of this work is to study the Expectation Propagation and Variational Inference methods from an intertwined viewpoints and a different divergence standpoint. We make a new unification of EP and VB algorithms which makes EP less analytically rigorous for the hierarchical Bayesian framework. Variational Bayes EP (VBEP) incorporates the propagation algorithm of EP at the VB updating stage. VBEP focuses on the core limitation of expectation propagation algorithm while generalizing the Variational inference algorithm. The most vital part of the EP and VB algorithm is the minimization of the Kullback-Leibler divergence. This new approach to both EP and VB method is called the Variational Bayes Expectation Propagation (VBEP) algorithm. VBEP has several advantages such as the generalization of the Variational Inference algorithm by incorporating the refining strategy of EP into VB. Moreover, the refinement of the prior through the data instead of fixed contribution of the prior to the approximate posterior in VB would be expected to improve the accuracy of VB although at the expense of the global update rule. Additionally, VBEP connect a path or mediates between EP and VB algorithms. VBEP breaches the rigorous analytical problems of EP by transiting from the VB algorithm with an augmentation. Additional, solving the intractable tilted posterior distribution of EP with a VB approach. This leads to working in the perspective of hybridizing the EP and VB to approximate hierarchical Bayesian models. We provide some theoretical framework that establish the connecting linkage between EP and VB.

Furthermore, we investigate the sparse linear models into nonlinear regression following the strategy by Jylanki et al. (2014) which combines the sparsity favoring priors with a two-layer regression models. This aims to solve the challenges faced in constructing a reliable Gaussian EP approximation for the analytically intractable likelihood resulting from the NN observation model by adopting the probabilistic backpropagation method by Jose & Ryan (2015).

Finally, we derive the VBEP for the hierarchical Bayesian model with the Logistic regression as the likelihood and the Gaussian prior distribution. We work in the context of deep neural networks. Working with Logistic regression by adopting the approximate lower bound of the logistic function [Jaakkola & Jordan (2000)] extends the work by Jose & Ryan (2015). However, this work computes the parameters of the hyperprior distribution using the marginal likelihood. This leads us to another useful algorithms such as VBEM, EP-EM and a new algorithm VEP-EM we would compare with. conventionally, following the widely used approximation method in Expectation Maximization algorithm, these algorithms compute the approximate posterior distribution at the E- step and optimize the parameters of the hyperprior distribution at the M-step. This is iterated until convergence.

A.1.2 The Heart of EP through VB

According to Minka (2005), EP minimizes the inclusive Kullback-Leibler divergence $\mathrm{KL}(p||q)$ that uses the matching-moment if only the two distributions are in the same exponential family most importantly Gaussian family. However, many studies have brought into limelight how rigorous and intractable EP could be when the tilted and approximate distribution come from different family of distributions entirely. Zoeter & Heskes (2005) uses Gaussian Quadrature for the problem of mismatching moments in EP with different family of distributions such as Beta distributions. Also, in Chapter 6 we have used stochastic search, and MCMC algorithm to solve the problem of mismatching moments between Gaussian and Exponential distributions. On the contrary, VB minimizes the exclusive Kullback-Leibler divergence $\mathrm{KL}(q||p)$. VB has been used to handle many combinations of incongruous distributions by indirectly maximizing the lower bound $\mathcal{L}(q)$ through optimizing with respect to the distribution q. The difference between these two Kullback-Leibler divergences can be understood by noting that there is a large positive contribution to the Kullback-Leibler divergence $\mathrm{KL}(q||p)$ from the region of the latent space in which the p is near zero unless q is also close to zero. Thus minimizing this form of KL divergence leads to distributions q that avoid the region in which p is small. Similar to this framework, Magnus et al. (2009) hybridized VB and EP for Bayesian sparse factor analysis. A comparison study

carried out by Kim & Wand (2016) on the accuracy power of the mean and variance estimates produced by both VB and EP algorithms shows that the mean estimate produced by VB tends to be more accurate than the mean estimate produced by EP algorithm. On the contrary, the variance estimate produced by VB is underestimated or less accurate than the variance estimate produced by EP algorithm. This also confirms the study by Bishop (2006) that shows how well approximated to the mean of the exact posterior distribution the mean of the approximate posterior produced by VB but underestimates the variance estimate.

Here, we show that the Kullback-Leibler divergence $\mathrm{KL}(p||q)$ is less than or equal to $\mathrm{KL}(q||p)$ augmented by any constant and local optimization to reflect the Leave-One-Out (LOO) method in EP. i.e.

$$\mathrm{KL}(p||q) \leq \mathrm{KL}(q||p) + \text{constant} \tag{A.1}$$

The accuracy of the pure mean-field solution, treating the latent variables as factorized variables by augmenting the exclusive KL divergence. By minimizing the $\mathrm{KL}(p||q)$ of Equation (A.1), the normalization constant Z following the matching moments of Minka (2001), the Kullback-Leibler divergence between p and q^{new} can then be obtained as a function of m, v, and the gradient of $\log Z$ with respect to these quantities, namely

$$\begin{aligned} m^{new} &= m_{-i} + v_{-i}\nabla_m \log Z \\ v^{new} &= v_{-i} - v_{-i}^2\left[\left(\nabla_m \log Z\right)^2 - 2\nabla_v \log Z\right] \end{aligned} \tag{A.2}$$

We present the VBEP for the Bayesian linear regression model example in Bishop (2006). This example has been solved by the variational Bayes approach. The likelihood function for $\mathbf{w}$, and the prior over $\mathbf{w}$ are given by

$$p(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N}\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1}) \text{ and } p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}) \tag{A.3}$$

where $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\mathbf{x}_i)$. The prior over α is given thus

$$p(\alpha) = Gam(\alpha|a_0, b_0) \tag{A.4}$$

First by using EP algorithm, we approach EP from $\mathrm{KL}(\hat{p}_i\|q_i)$ and $\mathrm{KL}(q_i||\hat{p}_i)$. Here as EP, we have the cavity distribution, approximate posterior, and the tilted posterior as follows

$$q_{-i}(\mathbf{w}) = \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i}),\ q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{m}_w, \boldsymbol{v}_w),\ \text{and}\ \hat{p}_i(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\hat{m}_i, \hat{v}_i) \tag{A.5}$$

we show that $\mathrm{KL}(\hat{p}_i\|q_i) \equiv \mathrm{KL}(q_i\|\hat{p}_i)$. However, it is clear from the symmetric property of Kullback-Leibler divergence that $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$. We proceed from the exclusive KL divergence first according to the conventional variational Bayes.

$$\ln q_i(\mathbf{w}) = E_\alpha\Big[\ln q_{-i}(\mathbf{w}) t_i(\mathbf{w})\Big] = E_\alpha\Big[\ln \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i})\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1})\Big] \tag{A.6}$$

We note here that the expectation with respect to $q(\alpha)$ is constant and it becomes irrelevant which will be removed going forward. Now, the mean and the variance of $q(\mathbf{w})$ is as follows

$$m_w = \Big(m_{-i} + v_{-i}\boldsymbol{\phi}_i t_i \beta\Big)\Big(1 + v_{-i}\boldsymbol{\phi}^T\boldsymbol{\phi}\beta\Big)^{-1} \text{ and } v_w = v_{-i}\Big(1 + v_{-i}\boldsymbol{\phi}^T\boldsymbol{\phi}\beta\Big)^{-1} \tag{A.7}$$

Now from the direction of inclusive Kullback-Leibler divergence, first we compute the normalizing constant

$$Z_t = \int \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i})\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1})\, dw = \mathcal{N}(m_t, v_t) \tag{A.8}$$

where the mean m_t and variance v_t of t are as follows

$$m_t = m_{-i}\boldsymbol{\phi}_i \text{ and } v_t = \Big(\beta^{-1} + v_{-1}\boldsymbol{\phi}^T\boldsymbol{\phi}\Big)^{-1} \tag{A.9}$$

Computing $\nabla_m \log Z_t$ and $\nabla_v \log Z_t$ and using Equation A.8 gives the following

$$m_w = \left(m_{-i} + v_{-i}\boldsymbol{\phi}_i t_i \beta\right)\left(1 + v_{-i}\boldsymbol{\phi}^T\boldsymbol{\phi}\beta\right)^{-1} \text{ and } v_w = v_{-i}\left(1 + v_{-i}\boldsymbol{\phi}^T\boldsymbol{\phi}\beta\right)^{-1} \quad \text{(A.10)}$$

This establishes the equivalence between $\mathrm{KL}(\hat{p}_i \| q_i) \equiv \mathrm{KL}(q_i \| \hat{p}_i)$ in a local approximation.

A.2 The Model

This section focuses on the multilayer perceptron NNs where the unknown function value $f_i = f(\mathbf{x}_i)$ related to a d- dimensional input vector $\mathbf{x}_i$ is modeled as

$$\hat{f}(\mathbf{x}_i) = \sum_{k=1}^{K} \mathbf{W}_{kL}^T g(\mathbf{W}_{kl}^T \boldsymbol{z}_{l-1}), \qquad l = 1, ..., L \quad \text{(A.11)}$$

where $g(x)$ is a nonlinear activation function, K the number of hidden units, and the $\mathcal{W} = \{\mathbf{W}_l\}_{l=1}^L$ is the collection or array of all the weights of the networks with dimension of $K_l \times (K_{l-1}+1)$ between the fully connected layers. We denote the output of the layers by by vectors $\{\boldsymbol{z}_l\}_{l=0}^L$ where $\boldsymbol{z}_0$ is the input layer. $\{\boldsymbol{z}_l\}_{l=1}^{L-1}$ represents the output of the hidden layer and $\boldsymbol{z}_L = \sigma(\hat{f}_i)$ is the output of the output layer. The activation functions for each hidden layer are Rectified Linear Units (RELUs) i.e., $a(x) = \max(x, 0)$, [Nair & Hinton (2010)]. In the next subsection, we explain the likelihood function for the model.

A.2.1 Likelihood Definitions

Here, we illustrate the use of local variational methods for the Bayesian logistic regression model. This focuses on the variational treatment based on the approach of Jaakkola & Jordan (2000). The variational treatment leads to the Gaussian approximation like the Laplace method. However, compared to the Laplace method, the greater flexibility of the variational approximation leads to improved accuracy. Furthermore, the variational approach can be as optimizing a well defined objective function given the rigorous bound on the model evidence.

A.2.2 Binary-Class Classification

Logistic regression has been treated from the standpoint of Monte Carlo sampling techniques [Dybowski & Roberts (2005)]. The output of the last layer is transformed using the sigmoid function for a binary output and softmax as a multiclass output. The variational approximation based on the lower bound allows the likelihood function for logistic regression, which is governed by the sigmoid or softmax to be approximated by the exponential of a quadratic form.

We first note that the conditional distribution for y can be written as

$$p(y_i|a_L, \boldsymbol{\Theta}) = \sigma(a_L)^{y_i}(1-\sigma(a_L))^{1-y_i}$$

$$= e^{-y_i a_L}\frac{e^{-a_L}}{1+e^{-a_L}} = e^{-y_i a_L}\sigma(-a_L) \tag{A.12}$$

where $a_L = \hat{f}_i$ and the $\boldsymbol{\Theta}$ is the collection of all the hyperparameters and $\hat{f}$ is from the equation A.11. The variational lower bound on the logistic sigmoid function in A.12 is given by

$$\sigma(u) \geq \sigma(\zeta)\exp\{(u-\zeta)/2 - \lambda(\zeta)(u^2-\zeta^2)\} \tag{A.13}$$

where $\lambda(\zeta) = \frac{1}{2\zeta}\left[\sigma(\zeta) - \frac{1}{2}\right]$. Therefore, the likelihood function is written as

$$p(y_i|a_L, \boldsymbol{\Theta}) = e^{y_i a_L}\sigma(-a_L) \geq e^{y_i a_L}\sigma(\zeta)\exp\{-(a_L+\zeta)/2 - \lambda(\zeta)((a_L)^2-\zeta^2)\} \tag{A.14}$$

Moreover, the bound is applied to each of the terms in the likelihood function separately, then there is a variational parameter ζ_i associated to each training set $(\mathbf{x}_i, y_i)$. Finally, the lower bound for the likelihood function will be denoted as

$$h(\boldsymbol{\Theta}, \boldsymbol{\zeta}) = \prod_{i=1}^{N}\sigma(\zeta_i)\exp\left\{y_i a_L - (a_L+\zeta_i)/2 - \lambda(\zeta_i)(a_L^2-\zeta_i^2)\right\} \tag{A.15}$$

The likelihood used is the lower bound of the Sigmoid function for binary classification which is presented in Equation A.15 and this makes the posterior analytically intractable.

A.2.3 Prior Definitions

We use the sparsity-promoting priors $p(w_{kjl}|\tau_{kjl})$ with hierarchical scale parameters τ_{kjl}^{-1} where the weight w_{kjl} is the k:th row and j:th column of the $\mathbf{W}_l$, τ_{kjl}^{-1} controls the prior variance of all the weights w_{kjl}. We place a Gaussian prior over the weights as follows

$$p(w_{kjl}|\tau_{kjl}) = \mathcal{N}(w_{kjl}|0, \tau_{kjl}^{-1}) \tag{A.16}$$

where the variance is τ_{kjl}^{-1} in equation A.16. The grouping of the weights can be chosen freely and also other weight prior distribution can be used in place of Gaussian distribution. The approximate inference on the variance parameters $\boldsymbol{\tau}_l^{-1} > 0$ is carried out using non-negative supported prior distribution to constrain the variance to be non-negative. In doing so, the computationally most convenient alternative non-negative supported prior distribution is to employ rectified Gaussian prior distribution on the precision controlling parameter as follows;

$$p(\tau_{kjl}) = \frac{2}{\text{erfc}(-m_0/\sqrt{v_0})}\mathcal{N}(\tau_{kjl}|m_0, v_0)U(\tau_{kjl}) \tag{A.17}$$

where $k = 1, ..., K$, $j = 1, ..., K_{l-1} + 1$ and $l = 1, ..., L$. Equation A.17 corresponds to a rectified-Gaussian prior for the associated layer prior precision $\boldsymbol{\tau}_l$ and $U(.)$ is a step function, m_0 and v_0 are the location and scale parameter for the precision, respectively. It is easy to see that the rectified Gaussian prior is conjugate to a Gaussian likelihood and the posterior can be computed in the same manner as the standard Gaussian distribution since the erfc(0) = 1, then we have constant of 2 in Equation A.17.

However to solve the EP algorithm, the rectified Gaussian prior is only computationally possible if the location parameter m_0 is fixed to zero, making the *erfc* function vanish. Also we note that the biases are already included in the setup of the weights matrices.

A.2.4 The Posterior Distribution

Given the previously explained prior definitions and a set of N observations $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{y} = [y_i, ..., y_N]^T$ and the features are $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$, the joint

posterior distribution of the prior and hyperparameters is as follows;

$$p(\mathbf{w}, \boldsymbol{\tau}|\mathcal{D}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) = Z^{-1}\prod_{i=1}^{N} h(y_i|\hat{f}_i, \zeta_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L} p(w_{kjl}|\tau_{kjl})\prod_{l=1}^{L} p(\boldsymbol{\tau}_l|\boldsymbol{\gamma}) \tag{A.18}$$

where the $\boldsymbol{\gamma} = \{\zeta, m_0, v_0\}$ contains all the hyperparameters to be computed at the E-step of the EM version of EP, VB, and VBEP algorithms, and Z_{EP} is the approximation of the marginal likelihood Z which is the marginal likelihood of the observations conditioned on $\boldsymbol{\gamma}$ as follows

$$Z = p(\mathbf{y}|\mathbf{X}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) = \int h(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) p(\mathcal{W}|\boldsymbol{\tau}) p(\boldsymbol{\tau}|\boldsymbol{\gamma}) d\mathbf{w} d\boldsymbol{\tau} \tag{A.19}$$

A.3 Approximate Inference

In this section, we describe how approximate Bayesian inference on the unknown model parameters $\mathbf{w}, \boldsymbol{\tau}$, and $\boldsymbol{\zeta}$ can be done efficiently using the variants of EP. First, in section A.3.1, we describe how the posterior approximation is formed using the approximate term and in section A.5, we discuss the hybridization of the VB and EP algorithm suitable for determining their parameters.

A.3.1 The Approximate Posterior

We form the analytically tractable approximation for the exact posterior distribution. We approximate all the likelihood and prior terms with unnormalized Gaussian distribution where appropriate. The Gaussian distribution has become a common use of approximating family for the weights of neural network, due to its matching moments nature [Seeger (2008)]. However, we use the rectified Gaussian distribution for the prior distribution over the precision of the weights of the neural networks. This is important, as to place a nonzero constraint on the prior distribution. On the contrary, one could consider other exponential family distribution such as the gamma distribution for the weight precision parameter [Jose & Ryan (2015)]. We

approximate the exact posterior distribution in Equation A.18 as follows

$$p(\mathbf{w}, \boldsymbol{\tau}|\mathcal{D}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) = Z^{-1}\prod_{i=1}^{N} h(y_i|\hat{f}_i, \zeta_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L} p(w_{kjl}|\tau_{kjl})p(\tau_{kjl}|\gamma_{jkl}) \tag{A.20}$$

$$\approx Z_{EP}^{-1}\prod_{i=1}^{N}\tilde{Z}_{y,i}\tilde{t}(\hat{f}_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\tilde{Z}_{kjl}^{w}\tilde{t}(w_{kjl})\tilde{t}(\tau_{kjl}) \tag{A.21}$$

A.4 The Likelihood Term Approximations

The exact likelihood terms that depend on the weights $\mathbf{w}$ through $\tilde{f}_i$ according to the Equation A.21

$$h(y_i|a_L, \zeta_i) \approx \tilde{Z}_{y,i}\tilde{t}_i(a_L|\tilde{m}_i, \tilde{v}_i) = \tilde{Z}_{y,i}\mathcal{N}(a_L|\tilde{m}_i, \tilde{v}_i) \tag{A.22}$$

where $\tilde{Z}_{y,n} = \int \tilde{t}(a_L|\tilde{m}_i, \tilde{v}_i)\, da_L$ is a scalar scaling parameter or normalizing constant. Here, we have assumed that all the weights are incorporated in $\mathbf{w}$ for both the hidden layer and the output layer. Note that the notation $\mathcal{N}$ is used for a normalized Gaussian distribution. Notice that we are approximating the lower bound to the sigmoid as the likelihood function that is the probability distribution which normalizes over the binary targets y_i, by an un-normalized Gaussian distribution over the latent variables $\hat{f}_i$. This is important because we are interested in how the likelihood behaves as a function of the latent $\hat{f}_i$. On the contrary, this is somewhat different from the regression setting which uses Gaussian distribution as the likelihood function and as linear model for the output y_i which makes it a Gaussian distribution. We compute the posterior to investigate how the likelihood function behaves as a function of $\hat{f}_i$.

A.4.1 The prior Term Approximation

The prior terms of all the weights w_{ijl} for $i = 1, ..., K$, $j = 1, ..., K_{l-1} + 1$ and $l = 1, ..., L$ are approximated conventional by Gaussian distribution We have used particularly a factorized distribution, due to the structure of the prior distribution

over the precision of the weights,

$$p(w_{kjl}|\tau_{kjl}) \approx \tilde{Z}^w_{kjl}\tilde{t}(w_{kjl})\tilde{t}^\tau_{kjl}(\tau_{kjl}) \propto \mathcal{N}(w_{kjl}|\tilde{m}^w_{kjl}, \tilde{v}^w_{kjl})\mathcal{N}(\tau_{kjl}|\tilde{m}^\tau_{kjl}, \tilde{v}^\tau_{kjl}) \tag{A.23}$$

where a factorized site approximation with location and scale parameters $\tilde{m}^w_{kjl}$ and $\tilde{m}^\tau_{kjl}$, $\tilde{v}^w_{kjl}$, and $\tilde{v}^\tau_{kjl}$ are associated with the network weights and precision respectively. The approximation term for the τ_{kjl} is also assumed to be the rectified Gaussian distribution and any other exponential distribution could also be appropriate.

A.4.2 The Joint Posterior Approximate

The product of the independence local likelihood $\tilde{t}_i$ is

$$q(\hat{\boldsymbol{f}}) = \prod_{i=1}^{N}\tilde{Z}_{y,i}\mathcal{N}(\hat{f}_i|m_{\hat{f}}, v_{\hat{f}}) = \mathcal{N}(\hat{\boldsymbol{f}}|\boldsymbol{m}_{\hat{f}}, \boldsymbol{v}_{\hat{f}})\prod_{i=1}^{N}\tilde{Z}_{y,i} \tag{A.24}$$

The prior and hyperprior that need to be processed multiple times using the expectation propagation are the factors in equation A.21 as follows

$$q(\mathbf{w}, \boldsymbol{\tau}) = \prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\tilde{Z}^w_{kjl}\mathcal{N}(w_{kjl}|m^w_{kjl}, v^w_{kjl})\mathcal{N}(\tau_{kjl}|m^\tau_{kjl}, v^\tau_{kjl}) \tag{A.25}$$

In Equation A.25, we use the assumption of independence between the approximate posterior distributions and use the method of factorized distribution as follows

$$q(\mathbf{w}, \boldsymbol{\tau}) = q(\mathbf{w})q(\boldsymbol{\tau}) \tag{A.26}$$

where the approximate posterior distribution for the network weights is given as follows

$$q(\mathbf{w}) = \prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\mathcal{N}(w_{kjl}|m^w_{kjl}, v^w_{kjl}) \tag{A.27}$$

Conceptually, one can think of the approximate posterior distribution for the hyperprior τ in two ways, either by combining the approximate terms and the hyperprior

distribution which gives the following

$$q(\boldsymbol{\tau}) = \prod_{k=1}^{K_l} \prod_{j=1}^{K_{l-1}+1} \prod_{l=1}^{L} \mathcal{N}(\tau_{kjl} | m_{kjl}^{\tau}, v_{kjl}^{\tau}) \tag{A.28}$$

Now, multiplying the parameters approximation of $\mathbf{w}$ and $\boldsymbol{\tau}$ together with the prior in Equation A.27 and A.28 give the approximate posterior

$$q(\mathbf{w}) \propto \mathcal{N}(\boldsymbol{M}_w, \boldsymbol{V}_w) \quad \text{and} \quad q(\boldsymbol{\tau}) \propto \mathcal{N}(\boldsymbol{M}_\tau, \boldsymbol{V}_\tau) U(\tau) \tag{A.29}$$

where using the Gaussian multiplication strategy gives

$$\boldsymbol{M}_\tau = \boldsymbol{V}_\tau \tilde{\boldsymbol{V}}_\tau^{-1} \tilde{\boldsymbol{M}}_\tau \quad \text{and} \quad \boldsymbol{V}_\tau = \left(\boldsymbol{V}_0^{-1} + \tilde{\boldsymbol{V}}_\tau^{-1} \right)^{-1} \tag{A.30}$$

where the marginal posterior for the precision τ_{kjl} is given by

$$q(w_{kjl}) \propto \mathcal{N}(m_{kjl}^{w}, v_{kjl}^{w}) \quad \text{and} \quad q(\tau_{kjl}) \propto \mathcal{N}(m_{kjl}^{\tau}, v_{kjl}^{\tau}) U(\tau) \tag{A.31}$$

where m_{kjl}^{w} and v_{kjl}^{w} are the mean and variance parameters for the approximate distribution of the network weights $q(w_{kjl})$ while m_{kjl}^{τ} and v_{kjl}^{τ} are the mean and variance parameters for the approximate distribution of the precision $q(\tau_{kjl})$. The mean vector $\boldsymbol{M}_\tau$ of the approximate posterior is the vector of m_{kjl}^{τ} and the covariance of the approximate posterior $\boldsymbol{V}_\tau$ is diagonal with v_{kjl}^{τ} for the approximate posterior.

A.5 Hybridization of VB and EP

The parameters of the local site approximations that define the approximate posterior distribution are determined using the hybridization of VB and EP. In the following, we give general description of the EP update for the likelihood and the weight prior terms. Here, we consider a sequentially updated EP.

A.5.1 EP Update For the Hyperprior Terms

As noted above in Equation A.23, each of the exact prior factors is approximated by a corresponding approximation prior give by

$$t(w_{kjl}, \tau_{kjl}) = \mathcal{N}(w_{kjl}|0, \tau_{kjl}^{-1}) \quad \text{and} \quad \tilde{t}(\tau_{kjl}) = \mathcal{N}(\tau_{kjl}|\tilde{m}_{kjl}^{\tau}, \tilde{v}_{kjl}^{\tau}) \tag{A.32}$$

First, we initialize all the $\tilde{t}(\tau_{kjl})$ uniformly, that is, $\tilde{m}_{kjl}^{\tau} = 0$ and $\tilde{v}_{kjl}^{\tau} = \infty$. EP starts to incorporate all the exact prior factors $t(w_{kjl}, \tau_{kjl})$ into q in $K \times (K_{l-1} + 1) \times L$ times. Here, we are interested in the individual precision parameter for each weights. This is relevant because it shows how accurate the weight estimates are at the update for each unit of every layer. The only demerit of this approach is the memory inefficiency. However, we don't store each update in memory. The first time $t(w_{kjl}, \tau_{kjl})$ is incorporated into q, we update $\tilde{t}(\tau_{kjl})$ and q as follows:

$$\tilde{m}_{kjl}^{\tau} = 0 \quad \text{and} \quad \tilde{v}_{kjl}^{\tau} = v_0, \quad m_{kjl}^{\tau} = 0 \quad \text{and} \quad v_{kjl}^{\tau} = v_0 \tag{A.33}$$

where v_0 is the parameter of the rectified Gaussian hyperprior on τ. On subsequent iterations, we refine $\tilde{t}(\tau_{kjl})$ by first removing the approximate factor from the approximate posterior of τ to obtain the cavity distribution. This cavity distribution is computed as the fraction of the q and $\tilde{t}$. The cavity marginal distribution on τ_{kjl} is therefore

$$q_{-kjl}(\tau_{kjl}) = q(\tau_{kjl})\tilde{t}(\tau_{kjl})^{-1} = \mathcal{N}(\tau_{kjl}|m_{-kjl}^{\tau}, v_{-kjl}^{\tau}) \tag{A.34}$$

where m_{-kjl}^{τ} and v_{-kjl}^{τ} are as follows:

$$(v_{-kjl}^{\tau})^{-1} = (v_{kjl}^{\tau})^{-1} - (\tilde{v}_{kjl}^{\tau})^{-1}$$

$$m_{-kjl}^{\tau} = m_{kjl}^{\tau} + (\tilde{v}_{kjl}^{\tau})^{-1} v_{-kjl}^{\tau}(m_{kjl}^{\tau} - \tilde{m}_{kjl}^{\tau}) \tag{A.35}$$

The cavity for the marginal distribution of w_{kjl} is also

$$q_{-kjl}(w_{kjl}) = q(w_{kjl})\tilde{t}(w_{kjl})^{-1} = \mathcal{N}(w_{kjl}|m_{-kjl}^{w}, v_{-kjl}^{w}) \tag{A.36}$$

where m^w_{-kjl} and v^w_{-kjl} are as follows:

$$(v^w_{-kjl})^{-1} = (v^w_{kjl})^{-1} - (\tilde{v}^w_{kjl})^{-1}$$

$$m^w_{-kjl} = m^w_{kjl} + (\tilde{v}^w_{kjl})^{-1} v^w_{-kjl}(m^w_{kjl} - \tilde{m}^w_{kjl}) \tag{A.37}$$

A.5.2 Computing the Tilted for τ_{kjl}, and w_{kjl}

After incorporating all the prior factors, we compute the tilted posterior distribution $\hat{p}(\tau_{kjl})$. The tilted distribution is formed by combining the cavity with the exact prior term $t(\tau_{kjl})$:

$$\hat{p}(\tau_{kjl}) = \hat{Z}_w^{-1} q_{-kjl} t(\tau_{kjl}) p(\tau_{kjl}) = \mathcal{N}(\tau_{kjl}|\hat{m}^\tau_{kjl}, \hat{v}^\tau_{kjl}) \tag{A.38}$$

where the normalizing factor Z_w is given as follows

$$Z_w = \int t(\tau_{kjl}) p(\tau_{kjl}) q_{-kjl}\, d\tau_{kjl}$$

$$= \int \mathcal{N}(\tau_{kjl}|0, v_0^\tau)\mathcal{N}(\tau_{kjl}|m^\tau_{-kjl}, v^\tau_{-kjl}) U(\tau_{kjl})\, d\tau_{kjl} \tag{A.39}$$

We compute the $\log Z_w$ from the $\mathrm{KL}(q||p)$ of VB instead of a direct computation of Z by $\mathrm{KL}(p||q)$. We compute $\log Z_w$ from $\mathrm{KL}(q||p)$, with $\theta_{kjl} = (w_{kjl}, \tau_{kjl})$ as follows:

$$-\mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \frac{\hat{p}_{kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})}\, d\theta_{kjl} \tag{A.40}$$

$$= \int q_{kjl}(\theta_{kjl}) \left[\log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{Z_w} \right) - \log q_{kjl}(\theta_{kjl}) \right] d\theta_{kjl} \tag{A.41}$$

By rearranging Equation A.41 we obtain

$$-\mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} - \log Z_w \tag{A.42}$$

Making the $\log Z_w$ the subject of the formula and rearranging we obtain

$$\log Z_w = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} + \mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) \tag{A.43}$$

where

$$\mathcal{L}(\tau_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} \tag{A.44}$$

then using factorized method $q_{kjl}(\theta_{kjl}) = q_{kjl}(w_{kjl}, \tau_{kjl}) = q_{kjl}(w_{kjl}) q_{kjl}(\tau_{kjl})$

$$\log Z_w = \mathcal{L}(\theta_{kjl}) + \mathrm{KL}(q_{kjl} || \hat{p}_{kjl}) \tag{A.45}$$

A.5.3 The hyperprior parameters τ_{kjl}

Here, just like VB, we maximize the lower bound in equation A.45 and we take the expectation with respect to the $q(w_{jkl})$ using the following factorized method. Thus, minimizing Kullback-Leibler divergence is equivalent to maximizing the lower bound, we select all the exact distributions that depend on only τ_{kjl} and obtain a general expression for the optimal solution $q(\tau_{kjl})$ as follows

$$\ln q^*(\tau_{kjl}) = \mathrm{E}_w \Big[t(w_{kj}, \tau_{kjl}) p(\tau_{kjl}) q_{-kjl}(\tau) \Big] + \text{const} \tag{A.46}$$

$$m^{\tau}_{kjl} = v^{\tau}_{kjl} (v^{\tau}_{-kjl})^{-1} m^{\tau}_{-kjl} - \frac{1}{2} v^{\tau}_{kjl} (v^{w}_{kjl} + [m^{w}_{kjl}]^2) \tag{A.47}$$

where we have used the expectation with respect to q_w and $\mathrm{E}(w^2) = v_w + m_w^2$ and Equation A.47 becomes

$$m^{\tau}_{kjl} = \left[m^{\tau}_{-kjl} - \frac{1}{2} v^{\tau}_{-kjl} (v^{w}_{kjl} + [m^{w}_{kjl}]^2) \right] \frac{v_0}{v^{\tau}_{-kjl} + v_0}$$

$$v^{\tau}_{kjl} = \left(v_0^{-1} + (v^{\tau}_{-kjl})^{-1} \right)^{-1} \tag{A.48}$$

A.5.4 The Prior weights w_{kjl}

We compute the approximate posterior mean m_w and variance v_w according to the setup of variational expectation propagation. We use the expectation with respect to the approximate posterior $q(\tau_{kjl})$ and factorize all that depend only on the weights

w_{kjl} as follows

$$\ln q^*(w_{kjl}) = \mathrm{E}_\tau \left[t(w_{kjl}, \tau_{kjl}) q_{-kjl}(w_{kjl}) \right] + \text{const} \tag{A.49}$$

then the mean and variance of approximate posterior $q(w_{kjl})$ are computed as

$$m^w_{kjl} = m^w_{-kjl}(v^w_{-kjl})^{-1} v^w_{kjl} \quad \text{and} \quad v^w_{kjl} = \left[(v^w_{-kjl})^{-1} + m^\tau_{kjl} \right]^{-1} \tag{A.50}$$

In Equation A.50, $\mathrm{E}[\tau_{kjl}] = m^\tau_{kjl}$. Finally, we update the parameters of the approximate factor $\tilde{t}(\tau_{kjl})$ and $\tilde{t}(w_{kjl})$

$$\begin{aligned}
\tilde{v}^\tau_{kjl} &= \left[(v^\tau_{kjl})^{-1} - (v^\tau_{-kjl})^{-1} \right]^{-1} \\
\tilde{m}^\tau_{kjl} &= \tilde{v}^\tau_{kjl} \left[m^\tau_{kjl}(v^\tau_{kjl})^{-1} - m^\tau_{-kjl}(v^\tau_{-kjl})^{-1} \right] \qquad \text{(A.51)} \\
\tilde{v}^w_{kjl} &= \left[(v^w_{kjl})^{-1} - (v^w_{-kjl})^{-1} \right]^{-1} \\
\tilde{m}^w_{kjl} &= \tilde{v}^w_{kjl} \left[m^w_{kjl}(v^w_{kjl})^{-1} - m^w_{-kjl}(v^w_{-kjl})^{-1} \right] \qquad \text{(A.52)}
\end{aligned}$$

respectively.

A.5.5 EP Update For the Likelihood Terms

Here, we consider the procedures for updating the likelihood sites $\tilde{t}(w_{kjl})$ and approximate posterior $q(w_{kjl})$ defined in Equation A.22. The exact likelihood terms $p(y_i|f_i)$ is a Logistic regression model and approximated by the lower bound from a Taylor series $h(\boldsymbol{\Theta}, \boldsymbol{\zeta})$ which does not depend on the weight precision τ_{kjl}. The posterior approximations can be factorized as $q(z_{kjl}, a_{kjl}, w_{kjl})$ and the exact likelihood is $p(y_i|a_L) = t(a_L)$ where a_L is the matrix multiplication for the last layer. Now, we

compute the Z_y as follows

$$Z_y = \int h(y_i|a_L)\mathcal{N}(a_L|m_{kj}^{a_L}, v_{kj}^{a_L})\, da_L \tag{A.53}$$

$$= \exp\left\{ y_i a_L - (a_L - \zeta_i)/2 - \lambda(\zeta_i)(a_L^2 - \zeta_i^2) - \frac{1}{2v_{kj}^{a_L}}(a_L^2 - m_{kj}^{a_L})^2 \right\} \tag{A.54}$$

by integrating out the matrix multiplication a_L, we have

$$Z_y = \mathcal{N}(y_i|m_y, v_y) \tag{A.55}$$

where the mean m_y and v_y are as follows

$$m_y = \frac{1}{2} - \frac{m_{kj}^{a_L}}{v_{kj}^{a_L}} \text{ and } v_y = \frac{1}{\zeta_i}\left[\frac{1}{2} - \sigma(\zeta_i)\right] - \frac{1}{v_{kj}^{a_L}} \tag{A.56}$$

Note that we have made use of the

$$\lambda(\zeta_i) = \frac{1}{2\zeta_i}\left[\sigma(\zeta_i) - \frac{1}{2}\right]$$

The updated rule for the mean and variance of the approximate posterior of $q(f_i) = q(a_L)$ in Equation A.24 is given below

$$m_{new}^{a_L} = m_{\text{old}}^{a_L} + [m_y - y_i]v_y^{-1} \text{ and } v_{new}^{a_L} = v_{old}^{a_L} + v_y^{-1}[2m_{old}^{a_L}(y_i - m_y) - 1] \tag{A.57}$$

A.6 Probabilistic Back-propagation

In this section we describe a probabilistic back-propagation algorithm for this model. PBP does not use point estimates for the synaptic weights in the network, [Jose & Ryan (2015)]. Instead, it uses a collection of one-dimensional Gaussian, each one approximating the marginal posterior distribution of a different weight. PBP also has two phases equivalent to the ones of BP. In the first phase, the input data is propagated forward through the network. However, since the weights are now random, the activation produced in each layer are also random and result in (intractable) distri-

butions. PBP sequentially approximates each of these distributions with a collection of one-dimensional Gaussian that match their marginal means and variances. At the end of this phase, PBP computes, instead of the prediction error, the logarithm of the marginal probability of the target variable. In the second phase, the gradients of this quantity with respect to the means and variances of the approximate Gaussian posterior are propagated back using reverse-mode differentiation as in classic back-propagation. These derivatives are finally used to update the means and variances of the posterior approximation.

A.6.1 Derivation of the gradients

We derive the gradient of the gradient of the logarithm of the marginal likelihood, that is the $\log Z_y$ given in Equation A.53, with respect to the means and variance of the network weights in the Gaussian approximate posterior q. In PBP, the corresponding algorithm has two variables such as the means and variance for each neuron. The activation function used at each layer is the RELU activation and this becomes random since the weights are now random. The output of each layer is denoted as z_l and the matrix multiplication is denoted by a_l for $l = 1, ..., L$. The activation function used for the last layer is the sigmoid function $\sigma(.)$. We start by propagating forward through the network from the input layer to the last layer called output layer. Let us assume for the moment that we have $L = 3$ before the general concept of PBP.

A.6.2 The Forward Propagation

Consider a class of neural networks defined the function form

$$z_L = \frac{1}{1 + \exp[-a_L]} \tag{A.58}$$

where a_L stands the matrix multiplication of the last layer and it is explicitly written as follows

$$a_L = \boldsymbol{w}_L^T g_L \left[\sum_{l=1}^{L-1} \sum_{k=1}^{K} g_l \left(\sum_{j=1}^{J} w_{kjl} \boldsymbol{z}_{l-1} \right) \right] \tag{A.59}$$

$$\text{For Layer } L = 1, \ \mathrm{E}\Big[\mathbf{w}_1^T z_0\Big] \text{ and } m_{kj}^w m^{z_0} \tag{A.60}$$

A.6.3 The Backpropagation

For the last layer,

$$\frac{\partial Z_y}{\partial m_{kj}^{w_L}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} \frac{\partial m_{kj}^{a_L}}{\partial m_{kj}^{w_L}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} \frac{\partial v_{kj}^{a_L}}{\partial m_{kj}^{w_L}},$$

$$\frac{\partial Z_y}{\partial v_{kj}^{w_L}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} \frac{\partial m_{kj}^{a_L}}{\partial v_{kj}^{w_L}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} \frac{\partial v_{kj}^{a_L}}{\partial v_{kj}^{w_L}} \tag{A.61}$$

For the hidden layers

$$\frac{\partial Z_y}{\partial m_{kj}^{w_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} \frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{w_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{w_l}},$$

$$\frac{\partial Z_y}{\partial v_{kj}^{w_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{w_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{w_l}} \tag{A.62}$$

where

$$\frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_{l+1}}} \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_{l+1}}} \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}}, \tag{A.63}$$

$$\frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_{l+1}}} \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_{l+1}}} \frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} \tag{A.64}$$

$$\frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} = \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}} + \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}},$$

$$\frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} = \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} + \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} \tag{A.65}$$

$$\frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} = \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}} + \frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}},$$

$$\frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} = \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} + \frac{\partial v_{kj}^{a_{l+!}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} \tag{A.66}$$

The mean and variance of the output of the matrix multiplication at each level are defined as $m_{kj}^{a_l}$ and $v_{kj}^{a_l}$ respectively. Also, the mean and variance of the activation function which becomes the input of the next layer are defined as $m_{kj}^{z_l}$ and $v_{kj}^{a_l}$ respectively. First, the matrix multiplication is randomized following from Equation A.60

by computing the first and second moments as follows

$$E[a_2] = E[w_{kj}^2]E[z^2] = (\text{Var}(w) + (E[w])^2)(\text{Var}(z) + (E[z])^2) \tag{A.67}$$

The first and second moments are given below

$$m_{kj}^{a_l} = m_{kj}^{z_{l-1}} m_{kj}^{w_l} \tag{A.68}$$

$$v_{kj}^{a_l} = (m_{kj}^{z_{l-1}})^2 v_{kj}^{w_l} + v_{kl}^{z_{l-1}} (m_{kj}^{w_l})^2 + v_{kj}^{z_{l-1}} v_{kj}^{w_l} \tag{A.69}$$

The randomized RELU activation function is given as follows

$$m_{kj}^{z_l} = \mathbf{\Phi}(\alpha_{kj}) \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}} \gamma_{kj} \right] \tag{A.70}$$

$$v_{kj}^{z_l} = m_{kj}^{z_l} \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}} \gamma_{kj} \right] \mathbf{\Phi}(-\alpha_{kj}) + \mathbf{\Phi}(\alpha_{kj}) v_{kj}^{a_l} (1 - \gamma_{kj}^2 - \gamma_{kj}\alpha_{kj}) \tag{A.71}$$

where $\gamma_{kj} = \phi(\alpha_{kj})/\mathbf{\Phi}(\alpha_{kj})$, $\alpha_{kj} = m_{kj}^{a_l}/\sqrt{v_{kj}^{a_l}}$ with ϕ, and $\mathbf{\Phi}$ denote the standard Gaussian pdf and cdf respectively. The gradients starting from the normalizing constant Z_y in Equation A.53 are as follows

$$\frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} = [m_y - y_i] v_y^{-1} (v_{kj}^{a_L})^{-1} \tag{A.72}$$

$$\frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} = \frac{1}{(v_{kj}^{a_L})^2} \left[\frac{1}{2v_y^2} (y_i - m_y)^2 + \frac{m_{kj}^{a_L}}{v_y} (y_i - m_y) - \frac{1}{2v_y} \right] \tag{A.73}$$

where Equation A.53 is brought forward for convenience

$$m_y = \frac{1}{2} - m_{kj}^{a_L} (v_{kj}^{a_L})^{-1} \text{ and } v_y = \frac{1}{\zeta_i} \left[\frac{1}{2} - \sigma(\zeta_i) \right] - \frac{1}{v_{kj}^{a_L}} \tag{A.74}$$

This is a gradient of the upper layer with respect to the lower layer of interest a_l and z_{l-1}

$$\frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{z_{l-1}}} = m_{kj}^{w_l}, \ \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{z_{l-1}}} = 0, \ \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{z_{l-1}}} = 2 m_{kj}^{z_{l-1}} v_{kj}^{w_l}, \text{ and } \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{z_{l-1}}} = (m_{kj}^{w_l})^2 + v_{kj}^{w_l}.$$

$$\text{and } \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{z_{l-1}}} = (m_{kj}^{w_l})^2 + v_{kj}^{w_l}. \tag{A.75}$$

This is a gradient with respect to the same layer of interest a_l and w_l

$$\frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{w_l}} = m_{kj}^{z_{l-1}}, \ \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{w_l}} = 0, \ \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{w_l}} = 2v_{kj}^{z_{l-1}} m_{kj}^{w_l}, \text{ and } \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{w_l}} = (m_{kj}^{z_{l-1}})^2 + v_{kj}^{z_{l-1}}. \tag{A.76}$$

This is a gradient with respect to the same layer of interest z_l and a_l

$$\frac{\partial m^{z_l}}{\partial m^{a_l}} = \mathbf{\Phi}(\alpha_{kj})\left[1 + \sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}}\right] + \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\phi(\alpha_{kj}) \tag{A.77}$$

$$\frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} = \frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\phi(\alpha_{kj}) + \mathbf{\Phi}(\alpha_{kj})\left[\sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \frac{\gamma_{kj}}{2\sqrt{v_{kj}^{a_l}}}\right] \tag{A.78}$$

$$\begin{aligned}\frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}} = {} & m_{kj}^{z_l}\left[1 + \sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}}\right]\mathbf{\Phi}(-\alpha_{kj}) + \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\phi(\alpha_{kj})v_{kj}^{a_l}(1 - \gamma_{kj}^2 - \alpha_{kj}\gamma_{kj}) \\ & - \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\left[m_{kj}^{z_l}\phi(\alpha_{kj})\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}} - \mathbf{\Phi}(-\alpha_{kj})\frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}}\right] \\ & - \mathbf{\Phi}(\alpha_{kj})v_{kj}^{a_l}\left[2\gamma_{kj}\frac{\partial \gamma_{kl}}{\partial m_{kj}^{a_l}} + \alpha_{kj}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}} + \gamma_{kj}\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\right]\end{aligned} \tag{A.79}$$

$$\begin{aligned}\frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} = {} & \mathbf{\Phi}(\alpha_{kj})\left\{\left[1 - \gamma_{kj}^2 - \alpha_{kj}\gamma_{kj}\right]\left[1 + v_{kj}^{a_l}\gamma_{kj}\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\right] - v_{kj}^{a_l}\left[2\gamma_{kj}\frac{\partial \gamma_{kl}}{\partial v_{kj}^{a_l}} + \alpha_{kj}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \gamma_{kj}\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\right]\right\} \\ & + m_{kj}^{z_l}\left\{\left[\sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \frac{\gamma_{kj}}{2\sqrt{v_{kj}^{a_l}}}\right]\mathbf{\Phi}(-\alpha_{kj}) - \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\right. \\ & \left.\left[\phi(\alpha_{kj})\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}} - \frac{\mathbf{\Phi}(-\alpha_{kj})}{m_{kj}^{z_l}}\frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}}\right]\right\}\end{aligned} \tag{A.80}$$

we now compute the γ and α with respect to $m_{kj}^{a_l}$ and $v_{kj}^{a_l}$

$$\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}} = -\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\left[\alpha_{kj}\gamma_{kj}(\alpha_{kj}) + \gamma_{kj}^2(\alpha_{kj})\right] \quad \text{and} \quad \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}} = \frac{1}{\sqrt{v_{kj}^{a_l}}} \tag{A.81}$$

$$\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} = -\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\left[\alpha_{kj}\gamma_{kj}(\alpha_{kj}) + \gamma_{kj}^2(\alpha_{kj})\right] \quad \text{and} \quad \frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}} = -\frac{m_{kj}^{a_l}}{2v_{kj}^{a_l}\sqrt{v_{kj}^{a_l}}} \tag{A.82}$$

Appendix B

Proof of Identifiability for MCWM

The proof is divided into two parts. The first part is built upon results given in Hennig (2000) while the second part is built upon the results given in Grün & Leisch (2008). Consider the class of models defined in Equation (4.2) and prove the equality as follows;

$$\sum_{g=1}^{G} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\beta}_j^g)\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g = \sum_{k=1}^{\tilde{G}} F(\boldsymbol{y}|\boldsymbol{x};\tilde{\boldsymbol{\beta}}_j^k)\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k \tag{B.1}$$

holds for almost all $\boldsymbol{x} \in \mathbb{R}^d$ and for all $\boldsymbol{y} \in \mathcal{Y}$ iff $G = \tilde{G}$ and there exists a one-to-one correspondence between $\{1, ..., G\}$ and $\{1, ..., \tilde{G}\}$ such that for each $g \in \{1, ..., G\}$ there exists a correspondent element $k \in \{1, ..., \tilde{G}\}$ and $\boldsymbol{\beta}_j^g = \tilde{\boldsymbol{\beta}}_j^k$, $\boldsymbol{\mu}_g = \tilde{\boldsymbol{\mu}}_k$, $\boldsymbol{\Sigma}_g = \tilde{\boldsymbol{\Sigma}}_k$, and $\pi_g = \tilde{\pi}_k$. Integrating both sides of Equation (B.1) over $\mathcal{Y}$ is as follows

$$\int_{\mathcal{Y}} \sum_{g=1}^{G} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\beta}_j^g)\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g \; d\boldsymbol{y} = \int_{\mathcal{Y}} \sum_{k=1}^{\tilde{G}} F(\boldsymbol{y}|\boldsymbol{x};\tilde{\boldsymbol{\beta}}_j^k)\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k \; d\boldsymbol{y} \tag{B.2}$$

$$\sum_{g=1}^{G} \phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g \int_{\mathcal{Y}} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\beta}_j^g) \; d\boldsymbol{y} = \sum_{k=1}^{\tilde{G}} \phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k \int_{\mathcal{Y}} F(\boldsymbol{y}|\boldsymbol{x};\tilde{\boldsymbol{\beta}}_j^k) \; d\boldsymbol{y} \tag{B.3}$$

Since

$$\int_{\mathcal{Y}} F(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\beta}_j^g) \; d\boldsymbol{y} = \int_{\mathcal{Y}} F(\boldsymbol{y}|\boldsymbol{x};\tilde{\boldsymbol{\beta}}_j^k) \; d\boldsymbol{y} = 1$$

then Equation (B.3) becomes

$$\sum_{g=1}^{G}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g=\sum_{k=1}^{\tilde{G}}\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k \tag{B.4}$$

Let us set $p(\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\pi})=\sum_{k=1}^{\tilde{G}}\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k$ and $p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\pi)=\sum_{k=1}^{G}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\pi_k$, where $(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})=\{(\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\pi_g);g=1,...,G\}$, $(\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})=\{(\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k,\tilde{\pi}_k);k=1,...,\tilde{G}\}$. Applying the Bayes' theorem gives $p(\mathcal{D}_g|\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})=\frac{\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k}{\sum_{t=1}^{\tilde{G}}\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_t,\tilde{\boldsymbol{\Sigma}}_t)\tilde{\pi}_t}$ and $p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})=\frac{\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g}{\sum_{s=1}^{G}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)\pi_s}$. Then Substituting Equation (4.15) and Equation (4.16) into Equation (B.1) and Equation (B.2) we get $p(\mathcal{D}_k|\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})=\frac{\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k}{p(\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})}$ and $p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})=\frac{\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g}{p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})}$ thus, Equation (B.1) can be written as

$$\begin{aligned}p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\Theta})&=p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})\sum_{g=1}^{G}F(\boldsymbol{y};\boldsymbol{M},\boldsymbol{\theta}^g)p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})\\&=p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\varphi})\end{aligned} \tag{B.5}$$

where

$$p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\varphi})=\sum_{g=1}^{G}F(\boldsymbol{y};\boldsymbol{M},\boldsymbol{\theta}^g)p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) \tag{B.6}$$

where also the positive weights $\gamma_g(\boldsymbol{x})$ in Equation (4.14) can be written as $\gamma_g(\boldsymbol{x})=p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})$. To complete the first part of the proof, since the $p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})$ and $p(\mathcal{D}_k|\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})$ are defined according to the Equation (B.3) and Equation (B.4), we get:

$$\pi_g=\int_{\mathcal{X}}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g\,d\boldsymbol{x}=\int_{\mathcal{X}}\frac{\phi_d(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\pi_g}{\sum_{s=1}^{G}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)\pi_s}\sum_{s=1}^{G}\phi_d(\boldsymbol{x};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)\pi_s\,d\boldsymbol{x}$$

since $p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) = p(\mathcal{D}_k|\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}})$, then

$$= \int_{\mathcal{X}} \frac{\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k)\tilde{\pi}_k}{\sum_{t=1}^{\tilde{G}}\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_t,\tilde{\boldsymbol{\Sigma}}_t)\tilde{\pi}_t} \sum_{t=1}^{\tilde{G}}\phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_t,\tilde{\boldsymbol{\Sigma}}_t)\tilde{\pi}_t \; d\boldsymbol{x} = \tilde{\pi}_k \int_{\mathcal{X}} \phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k) \; d\boldsymbol{x} \tag{B.7}$$

since $\int_{\mathcal{X}} \phi_d(\boldsymbol{x};\tilde{\boldsymbol{\mu}}_k,\tilde{\boldsymbol{\Sigma}}_k) \; d\boldsymbol{x} = 1$, then $\pi_g = \tilde{\pi}_k$. Following the same step, it can be seen that $\boldsymbol{\mu}_g = \tilde{\boldsymbol{\mu}}_k$ and $\boldsymbol{\Sigma}_g = \tilde{\boldsymbol{\Sigma}}_k$.

The class of models in Equation (B.6) is identifiable if the condition of intra-component label switching is fulfilled, this builds up the second part of the proof.

$$\sum_{g=1}^{G} p(\mathcal{D}_g|\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) \prod_i \prod_j F(\boldsymbol{y}_{ij};M_{ij},\boldsymbol{\theta}_{ij}^g)$$

$$= \sum_{k=1}^{\tilde{G}} p(\mathcal{D}_g|\boldsymbol{x};\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}},\tilde{\boldsymbol{\pi}}) \prod_i \prod_j F(\boldsymbol{y}_{ij};M_{ij},\tilde{\boldsymbol{\theta}}_{ij}^k) \tag{B.8}$$

implies that $G = \tilde{G}$ and there exists a one-to-one mapping between the two sets $\{1,...,G\}$ and $\{1,...,\tilde{G}\}$ such that $\boldsymbol{\theta}^g = \tilde{\boldsymbol{\theta}}^k$. Moreover the relationship between response variable $\boldsymbol{y}$ and the covariates $\boldsymbol{x}$ is $\boldsymbol{\theta}$.

$$\ln\left(\frac{\boldsymbol{\theta}_j}{\boldsymbol{\theta}_J}\right) = \boldsymbol{x}_i'\boldsymbol{\beta}_j \tag{B.9}$$

Exponentiating Equation (B.9) gives $e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j}$ and since $\boldsymbol{\theta}_j = \tilde{\boldsymbol{\theta}}_j$ then $\boldsymbol{\theta}_J = \tilde{\boldsymbol{\theta}}_J$ which implies that $e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j} = e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_j}$. Now following Grun and Leisch, **?**, we show that $e^{\boldsymbol{x}_i'(\boldsymbol{\beta}_j - \tilde{\boldsymbol{\beta}}_j)} = c_j$. Following Ingrassia et al. (2015), we introduce two sets

$$\mathcal{X} = \Bigg\{ \boldsymbol{x} \in \mathcal{R}^d : \text{ for each } g,w \in \{1,...,G\} \text{ and}$$

$$k,h \in \{1,...,\tilde{G}\} : e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j^g} = e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j^w} \implies \boldsymbol{\theta}_j^g = \boldsymbol{\theta}_j^w, \boldsymbol{\beta}_j^g = \boldsymbol{\beta}_j^w,$$

$$e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j^g} = e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_j^k} \implies \boldsymbol{\theta}_j^g = \tilde{\boldsymbol{\theta}}_j^k, \boldsymbol{\beta}_j^g = \tilde{\boldsymbol{\beta}}_j^k, e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_j^k} = e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_j^h} \implies \tilde{\boldsymbol{\theta}}_j^k = \tilde{\boldsymbol{\theta}}_j^h, \tilde{\boldsymbol{\beta}}_j^k = \tilde{\boldsymbol{\beta}}_j^h \Bigg\} \tag{B.10}$$

Since $\boldsymbol{\theta}^g \neq \boldsymbol{\theta}^w$ for $g \neq w$. The following holds for all $i \in \boldsymbol{I}_k$ and for $j = 1, ..., J-1$, $y_{ij} = \delta_{ij}$ and $y_{iJ} = M_{ij} - y_{ij}$, then Kronecker delta $\delta_{ij} = 1$ if $i = j$ and zero otherwise. The multinomial coefficients on both side of equation B.6 are canceled

$$e^{\boldsymbol{x}_i'(\boldsymbol{\beta}^g - \boldsymbol{\beta}^k)} = \frac{\sum_{g=1}^{G} p(\mathcal{D}_g|\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \left[e^{\boldsymbol{x}_i'\boldsymbol{\beta}^g} \prod_i \prod_j \left(\sum_{u=1}^{J} e^{\boldsymbol{x}_i'\boldsymbol{\beta}_u^g} \right)^{-M_{ij}} \right]}{\sum_{k=1}^{\tilde{G}} p(\mathcal{D}_k|\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \left[e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}^k} \prod_i \prod_j \left(\sum_{u=1}^{J} e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_u^k} \right)^{-M_{ij}} \right]} \tag{B.11}$$

for a fixed $\boldsymbol{x} \in \mathcal{X}$, according to Equation (B.4), $\gamma_1(\boldsymbol{x}), ..., \gamma_G(\boldsymbol{x})$ which is also $p(\mathcal{D}_1|\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}), .., p(\mathcal{D}_G|\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$ and $p(\mathcal{D}_1|\boldsymbol{x}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\pi}}), .., p(\mathcal{D}_{\tilde{G}}|\boldsymbol{x}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\pi}})$ which all sum to one. It follows that, for each $\boldsymbol{x} \in \mathcal{X}$, the density given in Equation (B.6) is then identifiable if and only $G = \tilde{G}$ and there exists $k \in \{1, ..., \tilde{G}\}$ such that $\boldsymbol{\theta}^g = \tilde{\boldsymbol{\theta}}^k$ and $p(\mathcal{D}_G|\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = p(\mathcal{D}_{\tilde{G}}|\boldsymbol{x}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\pi}})$, then the left hand side Equation (B.11) is constant for all $i \in \boldsymbol{I}_g$.

Appendix C

Derivation of Maximization via IRLS

The updated estimates $\psi^{(q+1)}$ are the solutions of the following M-step.

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{N} \sum_{g=1}^{G} z_{ig}^{(q)} \left\{ y_{i1} \ln \phi_{i1g} + \sum_{j=2}^{J} y_{ij} \ln \phi_{ijg} \right\} \tag{C.1}$$

Following from Equation (4.6) and $f(\boldsymbol{x}, \boldsymbol{\beta}_{jg}) = \boldsymbol{\beta}_{0jg} + \boldsymbol{x}'\boldsymbol{\beta}_{1jg}$, the update of $\boldsymbol{\psi}^{(q)}$ of Equation (C.1) is derived as follows

$$\boldsymbol{\beta}_{jg}^{(q+1)} = \boldsymbol{\beta}_{jg}^{(q)} + [I(\boldsymbol{\beta}_{jg}^{(q)})]^{-1} S(\boldsymbol{\beta}_{jg}^{(q)}) \tag{C.2}$$

where $I(\boldsymbol{\beta}_{jg}^{(q)})$ is the Fisher information matrix and $S(\boldsymbol{\beta}_{jg}^{(q)})$ is the score function. The parameters are estimated as follows:

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = \frac{\partial}{\partial \boldsymbol{\phi}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) \frac{\partial}{\partial f(\boldsymbol{x}; \boldsymbol{\beta})} \boldsymbol{\phi}(f) \frac{\partial}{\partial \boldsymbol{\beta}} f(\boldsymbol{x}; \boldsymbol{\beta}) \tag{C.3}$$

Maximizing Equation (C.1) with respect to $\boldsymbol{\beta}$ is equivalent to independently maximizing each J class and G component expressions

$$S(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} \left\{ \frac{y_{ij}}{\phi_{ijg}} - \frac{y_{i1}}{\phi_{i1g}} \right\} \tag{C.4}$$

using the expression $y_{i1} = n_i - y_{ij}$ and $\phi_{i1g} = 1 - \phi_{ijg}$ Equation (C.4) can be written as

$$S(\boldsymbol{\beta}_{jg}^{(q)}) = \frac{\partial}{\partial \phi_{ijg}} Q_1(\boldsymbol{\Omega}; \boldsymbol{\psi}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} \left\{ \frac{y_{ij}}{\phi_{ijg}} - \frac{n_i - y_{ij}}{1 - \phi_{ijg}} \right\} \frac{\partial}{\partial \boldsymbol{\beta}_{jg}} \boldsymbol{\phi}_{ijg} \tag{C.5}$$

The next equation will be derived on element-by-element basis, that is;

$$\frac{\partial}{\partial f_{ijg}(\boldsymbol{x}; \boldsymbol{\beta})} \boldsymbol{\phi}(f_{ijg}) = \frac{\partial}{\partial f_{ijg}(\boldsymbol{x}; \boldsymbol{\beta})} \frac{\exp\{f_{ijg}(\boldsymbol{x}; \boldsymbol{\beta}_{jg})\}}{\sum_j^J \exp\{f_{ijg}(\boldsymbol{x}; \boldsymbol{\beta}_{jg})\}} \tag{C.6}$$

$$= \frac{\exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\} \sum_{j=1}^{J} \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\} - \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\} \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}}{\left(\sum_{j=1}^{J} \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}\right)^2} \tag{C.7}$$

then we have,

$$= \frac{\exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}}{\sum_{j=1}^{J} \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}} \left(1 - \frac{\exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}}{\sum_{j=1}^{J} \exp\left\{f_{ijg}\big(\boldsymbol{x}; \boldsymbol{\beta}_{jg}\big)\right\}}\right) \tag{C.8}$$

Equation (C.6) is $\phi_{ijg}(1 - \phi_{ijg})\boldsymbol{x}_i$. The score function becomes

$$S(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i \phi_{ijg}^{(q)} (1 - \phi_{ijg}^{(q)}) \left\{ \frac{y_{ij}}{\phi_{ijg}^{(q)}} - \frac{y_{i1}}{\phi_{i1g}^{(q)}} \right\}. \tag{C.9}$$

Now, we derive the Fisher information matrix as follows;

$$S(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i \phi_{ijg}^{(q)} \phi_{i1g}^{(q)} \left\{ \frac{y_{ij}}{\phi_{ijg}^{(q)}} - \frac{y_{i1}}{\phi_{i1g}^{(q)}} \right\} \tag{C.10}$$

$$S(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i \left\{ y_{ij} \phi_{i1g}^{(q)} - y_{i1} \phi_{ijg}^{(q)} \right\} \tag{C.11}$$

using the expression $y_{i1} = n_i - y_{ij}$ and $\phi_{i1g} = 1 - \phi_{ijg}$ again Equation (C.11) becomes

$$= \sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i \left\{ y_{ij} - n_i \phi_{ijg}^{(q)} \right\} \tag{C.12}$$

$$- \frac{\partial}{\partial \beta_{jg}} S(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i \phi_{ijg}^{(q)} (1 - \phi_{ijg}^{(q)}) \boldsymbol{x}_i \tag{C.13}$$

$$I(\boldsymbol{\beta}_{jg}^{(q)}) = \sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i' v_{ijg} \boldsymbol{x}_i \tag{C.14}$$

The updated estimate is

$$\beta_{jg}^{(q+1)} = \beta_{jg}^{(q)} + \left(\sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i' v_{ijg} \boldsymbol{x}_i \right)^{-1} \left(\sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i v_{ijg} \zeta_{ij}^{*} \right) \tag{C.15}$$

$$\beta_{jg}^{(q+1)} = \left(\sum_{i=1}^{N} z_{ig}^{(q)} n_i \boldsymbol{x}_i' v_{ijg} \boldsymbol{x}_i \right)^{-1} \left(\sum_{i=1}^{N} z_{ig}^{(q)} \boldsymbol{x}_i v_{ijg} \zeta_{ij}^{(q)} \right) \tag{C.16}$$

where $v_{ijg} = \phi_{ijg}^{(q)}(1 - \phi_{ijg}^{(q)})$, $\zeta_{ij}^{(q)} = n_i \boldsymbol{x}_i \beta_{jg}^{(q)} + \zeta_{ij}^{*}$ and $\zeta_{ij}^{*} = y_{ij}/\phi_{ijg}^{(q)} - y_{i1}/\phi_{i1g}^{(q)}$. The weight v_{ijg} and the adjusted response $\zeta_{ij}^{(q)}$ are updated at each iteration based on the current estimates of the multinomial distribution probability ϕ_{ijg}.

Appendix D

Calculus of Variations

We can think of a function $y(x)$ as being a mapping that, for any input value x, returns an output value y. In the same spirit, *functional* $F[y]$ can be defined as an operator that takes a function $y(x)$ as its input and returns an output value F. In the field of machine learning, a commonly used functional is the entropy $\mathrm{H}[x]$ for a continuous variable x because for any choice of probability density function $p(x)$, it returns a scalar value representing the entropy of x under that density. Thus the entropy of $p(x)$ could be written as $\mathrm{H}[p]$.

A more common task in conventional calculus is to find a value of x that maximize or minimize a function $y(x)$. Similarly, the goal of calculus of variation is to seek a function $y(x)$ that either maximizes or minimizes a functional $F[y]$ according the task at hand. This means among all the possible functions $y(x)$, we wish to find the particular function for which the functional $F[y]$ is a maximum or minimum. The calculus of variations can be used, for example, to show that the maximum entropy distribution is a Gaussian.

We could evaluate a conventional derivative dy/dx by making an infinitesimal change ϵ to the variable x and then expanding in powers of ϵ, so that

$$y(x+\epsilon) = y(x) + \frac{dy}{dx}\epsilon + O(\epsilon^2) \tag{D.1}$$

and finally taking the limit $\epsilon \to 0$. Similarly, for a function in high-dimensional space, $y(x_i, ..., x_D)$, the corresponding partial derivatives are defined by

$$y(x_1 + \epsilon_1, ..., x_D + \epsilon_D) = y(x_1 + \epsilon_1, ..., x_D + \epsilon_D) + \sum_{d=1}^{D} \frac{\partial y}{\partial x_i} \epsilon_i + O(\epsilon^2). \tag{D.2}$$

The analogous definition of a functional derivative arises when we consider how a functional $F[y]$ changes with respect to a small change $\epsilon\zeta(x)$ of the function $y(x)$, where $\zeta(x)$ is an arbitrary function of x.

Let the functional derivative of $E[f]$ with respect to $f(x)$ be denoted by $\delta F/\delta f(x)$ which is defined as

$$F[y(x) + \epsilon\zeta(x)] = F[y(x)] + \epsilon \int \frac{\delta F}{\delta y(x)} \zeta(x) dx + O(\epsilon^2). \tag{D.3}$$

This is a natural extension of Equation (D.2) where $F[y]$ now depends on a continuous set of variables, called the values of y at all point x. One condition is that the functional must be stationary with respect to small variations in the function $y(x)$ gives

$$\int \frac{\delta E}{\delta y(x)} \zeta(x) dx = 0. \tag{D.4}$$

Equation D.4 must hold for an choice of $\zeta(x)$, and it follows that the functional derivative must also vanish. We also consider a functional that is defined by an integral over a function $\mathcal{H}(y, y', x)$ that depends on both function $y(x)$ and its derivative $y'(x)$ as well as having a direct depends on x

$$F[y] = \int \mathcal{H}(y(x), y'(x), x) dx \tag{D.5}$$

where the value of $y(x)$ is assumed to be fixed at the boundary of the region of integration. We now obtain the Equation (D.6) below if we consider the variations in the function $y(x)$

$$F[y(x) + \epsilon\zeta(x)] = F[y(x)] + \epsilon \int \left\{ \frac{\partial \mathcal{H}}{\partial y} \zeta(x) + \frac{\partial \mathcal{H}}{\partial y'} \zeta'(x) \right\} dx + O(\epsilon^2) \tag{D.6}$$

By integrating the second term we obtain the following

$$\int \zeta(x)\left\{\frac{\partial \mathcal{H}}{\partial y} + \frac{\partial \mathcal{H}}{\partial y'}\right\}dx = \int \frac{\partial \mathcal{H}}{\partial y}\zeta(x)dx - \int \frac{d}{dx}\left(\frac{\partial \mathcal{H}}{\partial y'}\right)\zeta(x)dx + \frac{\partial \mathcal{H}}{\partial y'}\zeta(x) \tag{D.7}$$

embedding the last term into $O(\epsilon^2)$, Equation (D.7) becomes

$$F[y(x) + \epsilon\zeta(x)] = F[y(x)] + \epsilon\int \zeta(x)\left\{\frac{\partial \mathcal{H}}{\partial y} - \frac{d}{dx}\left(\frac{\partial \mathcal{H}}{\partial y'}\right)\right\}dx + O(\epsilon^2). \tag{D.8}$$

The functional derivative is required to vanish and it gives

$$\frac{\partial \mathcal{H}}{\partial y} - \frac{d}{dx}\left(\frac{\partial \mathcal{H}}{\partial y'}\right) = 0 \tag{D.9}$$

which are also known as the Euler-Lagrangian equations.
Let's take an example. If

$$\mathcal{H} = y(x)^2 + (y(x))^2 \tag{D.10}$$

$\partial\mathcal{H}/\partial y = 2y(x)$ and $\partial\mathcal{H}/\partial y' = 2y'(x)$. Then the Euler-Lagrangian equations take the form

$$y(x) - \frac{d}{dx}(y'(x)) = y(x) - \frac{d^2y}{dx^2} = 0 \tag{D.11}$$

This second order differential equation can be solved for $y(x)$ by making use of the boundary conditions on $y(x)$. Optimizing a functional with respect to a probability distribution need the normalization constraint on the probabilities.

Appendix E

Alternating Direction Method of Multipliers

In chapter (6), we showed that the *alternating direction method of multipliers* (ADMM) is appropriate to distributed convex optimization, and in particular to large-scale problems arising from statistics and machine learning. ADMM takes the form of a decomposition-coordination procedure, in which the solutions to small local subproblems are coordinated to find a solution to a large global problem. ADMM can be viewed as a combination of dual decomposition [Everett (1963); Lasdon (1970); Geoffrion (1972); Luenberger (1973); Bensoussan et al. (1976)] and augmented Lagrangian methods [Hestenes (1969*a*); Hestenes (1969*b*); Miele, Cragg, Iver & Levy (1971); Miele, Cragg & Levy (1971); Miele et al. (1972)] for constrained optimization. ADMM integrates the decomposability of dual ascent [for more on dual ascent: Boyd & Vandenberghe (2004); Rockafellar (1970); Shor (1985)] with the superior convergence properties of the method of multipliers. Augmented Lagrangian methods The algorithm solves the problems in the form

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) + h(z) \\ \text{subject to} : \quad & Ax + By + Cz = d; \end{aligned} \tag{E.1}$$

with variables $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $z \in \mathbb{R}^q$ where $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, $C \in \mathbb{R}^{p \times q}$ and $d \in \mathbb{R}^p$. According to the purpose of this, we will assume that f, g and h are

convex functions. The optimal value of the problem in Equation (E.1) is denoted by

$$p^* = \inf\{f(x) + g(y) + h(z) | Ax + By + Cz = d\} \tag{E.2}$$

The augmented Lagrangian is formed as follows;

$$L_\rho = f(x) + g(y) + h(z) + \alpha^T(Ax + By + Cz - d) + (\rho/2)\Big|\Big|Ax + By + Cz - d\Big|\Big|_2^2 \tag{E.3}$$

ADMM consists of the iterations

$$x^{k+1} := \underset{x}{\operatorname{argmin}}\, L_\rho(x, y^k, z^k) \tag{E.4}$$

$$y^{k+1} := \underset{y}{\operatorname{argmin}}\, L_\rho(x^{k+1}, y, z^k) \tag{E.5}$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}}\, L_\rho(x^{k+1}, y^{k+1}, z, \alpha^k) \tag{E.6}$$

$$\alpha^{k+1} := \underset{\alpha}{\operatorname{argmin}}\, \alpha^k + \rho(Ax^{k+1} + By^{k+1} + Cz^{k+1} - d) \tag{E.7}$$

where $\rho > 0$. The algorithm is very similar to dual ascent and the method of multipliers: it consists of an x-minimization step in Equation (E.4), y-minimization step in Equation (E.5), z-minimization step in Equation (E.6), and the dual variable update in Equation (E.7) . The dual variable update uses the step size equal to the augmented Lagrangian parameter ρ similar to the method of multipliers.

The method of multipliers for Equation (E.1) has the form

$$(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) := \underset{x,y,z}{\operatorname{argmax}}\, L_\rho(x, y, z, \alpha^k)$$

$$\alpha^{k+1} = \alpha^k + \rho(Ax^{(k+1)} + By^{(k+1)} + Cz^{(k+1)} - d)$$

Contrary to augmented Lagrangian, the variables x, y, and z are updated in an alternating direction, which accounts for the term *alternating direction*.

Appendix F

Derivatives of SSEP and EP-ADMM

F.1 EP via Monte Carlo integration called SSEP

F.1.1 Incorporating the priors into q

We first incorporated priors on τ and λ as factors. These factors have the same functional form as

$$q(\mathcal{X}, \lambda, \tau) = \left[\prod_{i=1}^{m}\prod_{j=1}^{n}\mathcal{N}(\mathcal{X}_{ij}\Big|mx_{i,j}, vx_{i,j})\right]\exp(\lambda|\alpha_\lambda)\exp(\tau|\alpha_\tau) \tag{F.1}$$

The first update rules α_λ^{new} and α_τ^{new} for q is obtained by SSEP method which will be discussed in the algorithm below. The rest of the factors are sequentially incorporated into Equation (F.1) which is also updated in a similar manner. One difficulty encountered when applying the update rules in Minka (2001) is that the normalizer does not have a closed form. This brings about a uniqueness in SSEP which treats Z as follows;

$$\begin{aligned} Z &= \int \mathcal{N}(\mathcal{L}\mathcal{X}_{ij}\,|0,\,\tau)\,q(\mathcal{X}, \lambda, \tau)\,d\mathcal{X}\,d\lambda\,d\tau \\ &= \int \mathcal{N}(\mathcal{L}\mathcal{X}_{ij}\,|0,\,\tau)\,\exp(\tau\,|\,\alpha_\tau)\,d\mathcal{X}\,d\tau \end{aligned} \tag{F.2}$$

In Equation (F.2), the integral involves τ. We adopted the method by John et al. (2011). The normalizing factor Z in Equation (F.2) is solved for τ as follows,

$$Z_\tau = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathcal{L}\mathcal{X}_{ij}\,|\,0,\,\tau^{(k)})\,\exp(\tau^{(k)}\,|\,\alpha_\tau), \tag{F.3}$$

τ is generated from exponential distribution while $\mathcal{X}$ is initialized by $\mathcal{Y}$ to compute $\mathcal{L}\mathcal{X}$ which is in general the goal of this work. So the Monte Carlo integration of Equation (F.2) with respect to $\mathcal{X}$ becomes

$$Z_x = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathcal{Y}|\mathcal{G}\mathcal{X}_{ij},\,\lambda)\mathcal{N}(\mathcal{X}^{(k)}, mx_{ij}, vx_{ij}) \tag{F.4}$$

Here, $\mathcal{X}$ is generated from any distribution student-t, Normal distribution, or Uniform distribution from which $\mathcal{L}\mathcal{X}$ is recomputed. Here, we write the exact likelihood that needs to be processed multiple times i.e $N \times D$ times, where N and D represent the number of rows and columns respectively.

$$p(L\mathcal{X}|\tau) = \prod_{i=1}^{m}\prod_{j=1}^{n}\mathcal{N}(L\mathcal{X}_{ij}\big|0,\tau) \tag{F.5}$$

To approximate the posterior with SSEP algorithm, the posterior distribution can be factorized as follows

$$q(\mathcal{X},\lambda,\tau) = \left[\prod_{i=1}^{m}\prod_{j=1}^{n}\mathcal{N}(\mathcal{X}_{ij}\big|mx_{i,j}, vx_{i,j})\right]\exp(\lambda|\alpha_\lambda)\exp(\tau|\alpha_\tau) \tag{F.6}$$

The exact likelihood term is a joint distribution and is denoted as

$$t_{i,j}(\tau) = \mathcal{N}(L\mathcal{X}_{ij}\big|0,\tau) \tag{F.7}$$

Then, we approximate the exact likelihood term by choosing approximating term

$$\tilde{t}_{i,j}(\tau) = \exp(\tau|\tilde{\alpha}_{i,j}) \tag{F.8}$$

We choose term to refine by removing $\tilde{t}_{i,j}$ from q to compute the cavity distribution on $\mathcal{X}_{i,j}$ and τ

$$q_{-i,j}(\tau) = \exp(\tau|\alpha_{-i,j}) \tag{F.9}$$

$$\alpha_{-i,j} = \alpha_{i,j} - \tilde{\alpha}_{i,j} \tag{F.10}$$

After this, we perform the moment marching between $q(\tau)$ and a normalizing version of $t_{i,j}(\tau)q_{-i,j}(\tau)$. It is computed as follows

$$\underset{q}{\operatorname{argmin}}\, KL(\tilde{p}_{i,j}\Big|\Big|q_{i,j})$$

where

$$\tilde{p}_{i,j}(\tau) = \frac{t_{i,j}(\tau)q_{-i,j}(\tau)}{\int t_{i,j}(\tau)q_{-i,j}(\tau)d\tau} \tag{F.11}$$

Following Equation (F.11), the integral contains an intractable terms resulting from $\nabla_\omega \log Z$ where

$$Z = \int_\tau t_{i,j}(\tau)q_{-i,j}(\tau)d\tau \tag{F.12}$$

Our goal is to make a stochastic approximation of this gradient.

$$\nabla_\omega \log Z = \frac{1}{Z}\nabla_\omega \int_\tau t_{i,j}(\tau)q_{-i,j}(\tau)d\tau \tag{F.13}$$

$$\nabla_\omega \log Z = \frac{1}{Z}\int_\tau t_{i,j}(\tau)\nabla_\omega q_{-i,j}(\tau)d\tau \tag{F.14}$$

Equation (F.14) can be written as

$$\nabla_\omega \log Z = \frac{1}{Z}\int_\tau t_{i,j}(\tau)q_{-i,j}(\tau)\nabla_\omega \log q_{-i,j}(\tau)d\tau \tag{F.15}$$

referring to Equation (F.11) as the posterior, we have Equation (F.15) as an expectation

$$\nabla_\omega \log Z = E_{\hat{p}(\tau)}\Big[\nabla_\omega \log q_{-i,j}(\tau)\Big] \tag{F.16}$$

we use the identity $\nabla_\omega q_{-i,j}(\tau) = q_{-i,j}(\tau)\nabla_\omega \log q_{-i,j}(\tau)$. We can stochastically approximate the expectation using Monte Carlo integration as follows;

$$\nabla_\omega \log Z = \frac{1}{K}\sum_{k=1}^{K}\nabla_\omega \log q_{-i,j}(\tau) \tag{F.17}$$

Now, we substitute for $q_{-i,j}(\tau)$

$$\nabla_\alpha \log Z = \frac{1}{K}\sum_{k=1}^{K}\nabla_\alpha \log \Big(\exp(\tau|\alpha_{-i,j})\Big) \tag{F.18}$$

we compute the updates as follows;

$$\nabla_{\alpha_{-ij}} \log Z = \frac{1}{K}\sum_{k=1}^{K}\Big(\tau^{(k)}\Big) \tag{F.19}$$

The update rule is given in equation (F.20) below;

$$\alpha_\tau^{new} \approx \alpha_{-ij} + v_{\alpha_{-ij}}\zeta_{\alpha_{-ij}} \tag{F.20}$$

Where the mean $\frac{1}{K}\sum_{k=1}^{K}\tau^{(k)}$ is denoted as $\zeta_{\alpha_{-ij}}$ and variance as $v_{\alpha_{-ij}}$.

F.1.2 Incorporating the likelihood factors into q

Now, the $N \times D$ factors are sequentially incorporated for the likelihood in Equation (F.5). However, approaching this with conventional update rule in EP is difficult to compute because it requires integration of each likelihood factor. After incorporating all the factors in Equation (F.5), SSEP sequentially incorporates the $N \times D$ factors for the likelihood Equation (F.5) provided below in Equation (F.21)

$$\begin{aligned} Z_\lambda &= \int \mathcal{N}(\mathcal{Y}\,|\,\mathcal{G}\mathcal{X},\,\lambda)\,q(\mathcal{X},\lambda,\tau)\,d\mathcal{X}\,d\lambda\,d\tau \\ &= \int \mathcal{N}(\mathcal{Y}\,|\,\mathcal{G}\mathcal{X},\,\lambda)\,\exp(\lambda\,|\,\alpha_\lambda)d\lambda. \end{aligned} \tag{F.21}$$

Then the Monte Carlo approximation for Equation (F.21) is

$$\frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathcal{Y}|\mathcal{G}\mathcal{X},\lambda^{(k)})\exp(\lambda^{(k)}|\alpha_\lambda). \tag{F.22}$$

Now we compute the gradient of the normalizing factor as follows;

$$\nabla_{\alpha_\lambda}\log Z = \frac{1}{Z}\int_\lambda t_{i,j}(\lambda)\nabla_{\alpha_\lambda}q_{-i,j}(\lambda)d\lambda \tag{F.23}$$

$$\nabla_{\alpha_\lambda}\log Z = \frac{1}{Z}\int_\lambda t_{i,j}(\lambda)q_{-i,j}(\lambda)\nabla_{\alpha_\lambda}\log q_{-i,j}(\lambda)d\lambda \tag{F.24}$$

$$\nabla_{\alpha_\lambda}\log Z = \frac{1}{Z}\int_\lambda \mathcal{N}(\mathcal{Y}|\mathcal{G}\mathcal{X},\lambda)q_{-i,j}(\lambda)\nabla_{\alpha_\lambda}\log\exp(\lambda|\alpha_\lambda)d\lambda \tag{F.25}$$

Equation (F.25) can be approximated as follows;

$$E_{\hat{p}}[\lambda] = \sum_{k=1}^{K}\nabla_{\alpha_\lambda}\log\exp(\lambda|\alpha_\lambda) \tag{F.26}$$

Similarly, for λ

$$\nabla_\omega\log Z = \frac{1}{K}\sum_{k=1}^{K}\nabla_\omega\log\Big(\exp(\lambda|\alpha^\lambda_{-i,j})\Big) \tag{F.27}$$

$$\nabla_{\alpha^\lambda_{-ij}}\log Z = \frac{1}{K}\sum_{k=1}^{K}\Big(\lambda^{(k)}\Big) \tag{F.28}$$

We denote the mean $\frac{1}{K}\sum_{k=1}^{K}\lambda^{(k)}$ as $\zeta_{\lambda_{-ij}}$ and variance as $v_{\alpha^\lambda_{-ij}}$. The update rule for λ is given in Equation (F.29) below;

$$\alpha^{new}_\lambda \approx \alpha^\lambda_{-ij} + v_{\alpha^\lambda_{-ij}}\frac{1}{K}\sum_{k=1}^{K}\lambda^{(k)} \tag{F.29}$$

The cavity of the prior over $\mathcal{X}$ is

$$q_{-ij}(\mathcal{X}) = \mathcal{N}(x_{ij};m_{-ij},v_{-ij}) \tag{F.30}$$

We choose the approximate posterior for $\mathcal{X}$ as

$$q(\mathcal{X}) = \mathcal{N}(x_{ij}; m_x, v_x) \tag{F.31}$$

Then the approximate term is chosen from the Gaussian family

$$\tilde{t}_{ij}(\mathcal{X}) = \mathcal{N}(\mathcal{X}_{ij}; \tilde{m}_{ij}, \tilde{v}_{ij}) \tag{F.32}$$

In order to update the cavity distribution we remove $\tilde{t}_{ij}$ from q and update its parameters as follows;

$$v_{-i,j}^{-1} = vx_{i,j}^{-1} - \tilde{v}_{i,j}^{-1} \tag{F.33}$$

$$m_{-i,j} = mx_{i,j} + v_{i,j}\tilde{v}_{i,j}(mx_{i,j} - \tilde{m}_{i,j}) \tag{F.34}$$

Our goal is to make a stochastic approximation of this gradient.

$$\nabla_\omega \log Z = \frac{1}{Z}\nabla_\omega \int_{\mathcal{X}_{i,j},\tau} t_{i,j}(L\mathcal{X}_{i,j}, \tau) q_{-i,j}(\mathcal{X}_{i,j}, \tau) d\mathcal{X}_{i,j} d\tau \tag{F.35}$$

Replace $q_{-i,j}$ in Equation (F.35) with $q_{i,j}/\tilde{t}_{i,j}$ then it becomes

$$\nabla_\omega \log Z = \frac{1}{Z} \int_{\mathcal{X}_{i,j},\tau} \frac{t_{i,j}(L\mathcal{X}_{i,j}, \tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j}, \tau)} \nabla_\omega q_{i,j}(\mathcal{X}_{i,j}, \tau) d\mathcal{X}_{i,j} d\tau \tag{F.36}$$

$$\nabla_\omega \log Z = \frac{1}{Z} \int_{\mathcal{X}_{i,j},\tau} \frac{t_{i,j}(L\mathcal{X}_{i,j}, \tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j}, \tau)} q_{i,j}(\mathcal{X}_{i,j}, \tau) \nabla_\omega \log q_{i,j}(\mathcal{X}_{i,j}, \tau) d\mathcal{X}_{i,j} d\tau \tag{F.37}$$

At this juncture, we proceed with the use of Monte Carlo Integration

$$\frac{\delta_1}{\delta_0} = \sum_{n=1}^{N} \frac{t_{i,j}(L\mathcal{X}_{i,j}, \tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j}, \tau)} \nabla_\omega \log q_{i,j}(\mathcal{X}_{i,j}, \tau) \Big/ \sum_{n=1}^{N} \frac{t_{i,j}(L\mathcal{X}_{i,j}, \tau)}{\tilde{t}_{i,j}(\mathcal{X}_{i,j}, \tau)} \tag{F.38}$$

when taking $\mathcal{X}$ into consideration then we fix τ and vice versa. So we update the parameter by taking the gradient step

$$\omega^{t+1} = \omega^t + lr * \frac{\delta_1}{\delta_0} \tag{F.39}$$

F.2 Derivative of EP-ADMM in High dimensional space

We introduce the Alternating direction method of multiplier (ADMM) algorithm to update the approximate posterior parameters.

$$\begin{aligned} &\text{minimize} \quad \mathrm{KL}(\tilde{p}(x_{ij})||q(x)_{ij})) \\ &\text{subject to} : m_{ij} \geq a; v_{ij} \geq b \end{aligned} \tag{F.40}$$

where a and b are constants. Then updating according to Minka (2001) is as follows;

$$m_x = \nabla_{m_{-ij}} \log Z_x + \alpha + \rho(m_{-ij} - a) \tag{F.41}$$

and

$$v_x = \nabla_{v_{-ij}} \log Z_x + \beta + \rho(v_{-ij} - b) \tag{F.42}$$

where

$$q(x) = \mathcal{N}(\mathcal{X}; m_x, v_x) \tag{F.43}$$

then the tilted distribution is

$$\tilde{p}(x_{ij}) = \frac{t_{ij}(x) q_{-ij}(x)}{\int_{\mathcal{X}_{ij}} t_{ij}(x) q_{-ij}(x) dx_{ij}} \tag{F.44}$$

Now the normalizing factor is

$$Z_x = \int_{x_{ij}} \mathcal{N}(\mathcal{Y}_{ij}; G\mathcal{X}_{ij}, \lambda) \mathcal{N}(\mathcal{X}_{ij}; m_{-ij}, v_{-ij}) dx_{ij} \tag{F.45}$$

By integrating out x, Z_x becomes

$$\int \exp\left\{ -\frac{1}{2} \Big(y_{ij} - g x_{ij} \Big)^2 \lambda^{-1} + \Big(x_{ij} - m_{-ij} \Big)^2 v_{-ij}^{-1} \right\} dx_{ij} \tag{F.46}$$

Completing the squares, we have

$$-\frac{1}{2}\left[x_{ij}-\frac{gy_{ij}\lambda^{-1}+m_{-ij}v_{-ij}^{-1}}{g^2\lambda^{-1}+v_{-ij}^{-1}}\right]^2\left(g^2\lambda^{-1}+v_{-ij}^{-1}\right)+\frac{1}{2}\frac{\left(gy_{ij}\lambda^{-1}+m_{-ij}v_{-ij}^{-1}\right)^2}{g^2\lambda^{-1}+v_{-ij}^{-1}} \tag{F.47}$$

Then the normalizing factor of x is

$$Z_x=\mathcal{N}(y_{ij};m_y,v_y) \tag{F.48}$$

where

$$m_y=\frac{g\lambda^{-1}m_{-ij}v_{-ij}^{-1}}{\lambda^{-1}-g^2\lambda^{-2}\left(g^2\lambda^{-1}+v_{-ij}^{-1}\right)^{-1}}=gm_{-ij} \tag{F.49}$$

and

$$\frac{1}{\lambda^{-1}-g^2\lambda^{-2}\left(g^2\lambda^{-1}+v_{-ij}^{-1}\right)^{-1}}=v_{-ij}g^2+\lambda \tag{F.50}$$

then updating the parameters m_x and v_x of $q(\mathcal{X})$ as follows according to Minka (2001) with a method of multipliers, we have

$$m_x^{new}=m_{-ij}+v_{-ij}\frac{y_{ij}-gm_{-ij}}{v_{-ij}g^2+\lambda}g+\alpha+\rho(m_i-a) \tag{F.51}$$

$$v_x^{new}=\frac{v_{-ij}\lambda}{v_{-ij}g^2+\lambda}+\beta+\rho(v_i-b) \tag{F.52}$$

$$\alpha^{new}=\alpha^k+\rho(m_x^{new}-a)\quad\text{and}\quad\beta^{new}=\beta^k+\rho(v_x^{new}-b) \tag{F.53}$$

LIST OF ONGOING PUBLICATIONS

1. The Variants of Expectation Maximization algorithms for Scaling Multinomial Cluster-Weighted Models.

2. Image Reconstruction by Splitting Expectation Propagation Techniques from Iterative Inversion.

3. A Cluster Weighted Models based on T distributed stochastic Neighbor Embedding with application to Epileptic Seizure recognition data.

4. Zero-inflated Poisson Cluster Weighted Models for handling class imbalance in medical data.

5. Variational Bayes EP Algorithms with Probabilistic Backpropagation for Bayesian Neural Networks.

REFERENCES

(n.d.).

Aitkin, M. & Aitkin, I. (1994), 'Efficient computation of maximum likelihood estimates in mixture distributions, with reference to overdispersion and and variance components.', *In Proceedings XVIIth International Biometric Conference. Alexandria, Virginia: Biometric Society,* pp. 123–138.

Akaike, H. (1973), 'Information theory and an extension of the maximum likelihood principle.', *In Second International Symposium Information Theory, B.N. Petrov and F. Csaki (Eds.). Budapest: Akademiai Kaido, pp. 267–281. (Reproduced in (1992) in Breakthroughs in Statistics, Kotz and N.L. Johnson (Eds.). New York: Springer-Verlag, pp* **1**, 610–624.

Akaike, H. (1974), 'A new look at the statistical model identification.', *IEEE Transaction on Automatic Control.* **19**, 716–723.

Alfo, M. & Trovato, G. (2004), 'Semiparametric mixture models for multivariate count data, with application.', *Econmics Journal* **7**(2), 426–454.

Amari, S. I. (1982), 'Differential geometry of curved exponential families-curvatures and information loss', *The Annals of Statistics* p. 357–385.

Amari, S. I. (1985), 'Differential-geometrical methods in statistics', *Springer* pp. 1–10.

Amari, S. I. (2009), 'α-divergence is unique, belonging to both f-divergence and bregman divergence classes', *IEEE Transactions on Information Theory* **55**(11), 4925–4931.

Aykroyd, R. G. (2015), 'Industrial tomography: Systems and applications', *1st Edition, Woodhead Publishing, ISBN-10: 1782421181* **1**, 772.

Banfield, J. D. & Raftery, A. E. (1993), 'Model-based gaussian and non-gaussian clustering.', *Biometrics* **49**, 803–821.

Barvinok, A. I. (1999), 'Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor', *Random Structures and Algorithms* **14**(1), 29–61.

Baudry, J., Raftery, A., Celeux, G., Lo, K. & Gottardo, R. G. (2010), 'Combining mixture components for clustering.', *Journal of Computational and Graphical Statistics, to appear* pp. 1–10.

Beal, M. (2003), 'Variational algorithms for approximate bayesian inference.', *PhD thesis, Gatsby Computational Neuroscience Unit, University College London* pp. 1–10.

Bensoussan, A., Lions, J. L. & Temam, R. (1976), 'Sur les mthodes de dcomposition, de dcentralisation et de coordination et applications.', *Methodes Mathematiques de l'Informatique* pp. 133–257.

Berta, P., Ingrassia, S., Punzo, A. & Vittadini, G. (2016), 'Multilevel cluster-weighted models for the evaluation of hospitals.', *DOI 10.1007/s40300-016-0098-3* **2**(74), 275–292.

Biernacki, C., Celeux, G. & Govaert, G. (2000), 'Assessing a mixture model for clustering with the integrated complete likelihood.', *Pattern Analysis and Machine Intelligence.* **22**(7), 719–725.

Biernacki, C., Celeux, G. & Govaert, G. (2003), 'Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. computational statistics and data analysis', *Biometrics.* **413**, 561–575.

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

Blei, D. & Jordan, M. (2006), 'Variational inference for dirichlet process mixtures.', *Bayesian Analysis.* **1**, 121–144.

Blei, D., M., Kucukelbir, A. & McAuliffe, J. D. (2018), 'Variational inference: A review for statisticians', *arXiv.* pp. 1–41.

Blischke, W. R. (1964), 'Estimating the parameters of mixtures of binomial distributions.', *Journal of the American Statistical Association* **59**(306), 510–528.

Bohning, D., Schlattmann, P. & Lindsay, B. (1994), 'Recent developments in computer-assisted analysis of mixtures', *Biometrics* **54**, 525–536.

Bouveyron, C., Celeux, G., Murphy, T. B. & Raftery, A. E. (2019), 'Model-based clustering and classification for data science with application in r', *Cambridge University Press, United Kingdom* p. 21.

Bouveyron, C., Girard, S. & Schmid, C. (2007), 'High-dimensional data clustering.', *Computational Statistics and Data Analysis* **52**(1), 502–519.

Boyd, S. & Vandenberghe, L. (2004), 'Convex optimization.', *Cambridge University Press.* pp. 1–10.

Boyen, X. & Koller, D. (1998), 'Tractable inference for complex stochastic processes. *Uncertainty in Al.*', pp. 1–10.

Bozdogan, H. (1987), 'Model selection and aikaike's infromation criterion (aic): The general theory and its analytical extensions,', *Psychoetrika.* **52**, 345–370.

Bozdogan, H. (1994), 'Theory and methodology of time series analysis.', *In Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach.* **1**.

Brannas, K. & Rosenqvist, G. (1994), 'Semiparametric estimation of heterogeneous count data models.', *European Journal of Operating Research* **76**, 247–258.

Braun, M. & McAuliffe, J. (2010), 'Variational inference for large-scale models of discrete choice', *Journal of the American Statistical Association* **105**(489), 324–335.

Bregman, L. M. (1967), 'The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming', *USSR Computational Mathematics and Mathematical Physics* **7**(3).

Brooks, S. P. & Gelman, A. (1998), 'General methods for monitoring convergence of iterative simulations.', *Journal of Computational and Graphical Statistics* **7:4**(4), 434 – 455.

Butler, R. (1986), 'Predictive likelihood inference with applications (with discussion).', *Journal of the Royal Statistical Society* **48**, 1–38.

Celeux, G. & Govaert, G. (1995), 'Gaussian parsimonious clustering models.', *Pattern Recognition* **28**(25, 76, 171, 237, 248), 781–793.

Celeux, G. & Soromenho, G. (1996), 'An entropy criterion for assessing the number of clusters in a mixture model', *Classification Journal* **13**, 195–212.

Charlier, C. (1996), 'Researchers into the theory of probability.', *Lunds University Arskriftm Ny foljd* **21**(5), 107–127.

Chernoff, H. (1952), 'A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations.', *The Annals of Mathematical Statistics* p. 493–507.

Corana, A., Marchesi, M., Martini, C. & Ridella, S. (1987), 'Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm.', *Journal of ACM Transactions on Mathematical Software,* **13**(3), 262–280.

Core, R. (2019), 'R: A language and environment for statistical computing.', *Vienna, Austia: R Foundation for Statistical Computing.* p. 1.

Cortez, P. & Silva, A. (2008), 'Using data mining to predict secondary school student performance.', *In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008)* (5), 5–12.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. & Spiegelhalter, D. J. (1999), 'Probabilistic networks and expert systems', *Springer-Verlag, New York* pp. 1–10.

Cox, D. R. & Wermuth, N. (1992), 'Response models for mixed binary and quintitative variables.', *Biometrika* **79**, 441–461.

Cox, R. (1946), 'Probability, frequency, and reasonable expectation.', *American Journal of Physics* **1**(14), 1–13.

Csiszár, I. (1963), 'Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten', *Magyar. Tud. Akad. Mat. Kutató Int. Közl* **8**, 85–108.

Dasgupta, A. & Raftery, A. E. (1998), 'Detecting features in spatial point processes with clutter via model-based clustering.', *Journal of the American Statistical Association.* **93**, 294–302.

David, A. & Dennis, K. (1988), 'Instance-based prediction of heart-disease presence with the cleveland database.', *tech. rep., University of California.* pp. 1–10.

Day, N. (1969), 'Estimating the components of a mixture of two normal distributions.', *Biometrika* **56**, 463–474.

Dayton, C. & Macready, G. (1988), 'Concomitant-variables latent-class models.', *Journal of the American Statistical Association* **401**(83), 173–178.

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

DeSarbo, W. & Cron, W. (1988), 'A maximum likelihood methodology for clusterwise linear regression.', *Journal of Classification* **5**, 249–282.

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S. & Froelicher, V. (1989), 'International application of a new probability algorithm for the diagnosis of coronary artery disease.', *Amerian Journal of Cardiology.* **64**, 304–310.

Dua, D. & Graff, C. (2017), 'Uci, machine leaning repository', *University of California, Irvine, School of Information and Computer Sciences* pp. 1–10.

Duda, R. & Hart, P. (1973), 'Pattern classification and scene analysis.', *New York: Wiley* pp. 1–10.

Dybowski, R. & Roberts, S. (2005), 'An anthology of probabilistic models for medical informatics. in d. husmeier, r. dybowski, and s. roberts (eds.)', *Probabilistic Modeling in Bioinformatics and Medical Informatics* **1**, 297–349.

Efron, B. (1975), 'Defining the curvature of a statistical problem (with applications to second order efficiency)', *The Annals of Statistics* pp. 1189–1242.

Efron, B. (1978), 'The geometry of exponential families', *The Annals of Statistics* **2**(6), 362–376.

Egan, J. (1975), 'Signal detection theory and roc analysis series in cognition and perception,', *Academic Press, New York.* pp. 1–10.

Elmore, R. T. & S., W. (2003), 'Identifiability and estimation in finite mixture models with multinomial components.', *Technical Report 03-04, Department of Statistics, Pennsylvania State University* **1**, 510–528.

Escobar, M. & West, M. (1995), 'Bayesian density estimation and inference using mixtures.', *Journal of the Americam Statistical Association* **90**, 577–588.

Everett, H. (1963), 'Generalized lagrange multiplier method for solving problems of optimum allocation of resources.', *Operations Research* pp. 399–417.

Everitt, B. (1996), 'An introduction to finite mixture distributions.', *Statistical Methods in Medical Research* **5**, 107–127.

Feynman, R. P., Leighton, R. B. & Sands, M. (1964), 'The feynman lectures of physics.', *Addison-Wesley* **2**(Chapter 19).

Figueiredo, M. A. T. & Jain, A. K. (2002), 'Unsupervised learning of finite mixture models.', *Transactions on Pattern Analysis and Machine Intelligence.* **24**(3), 381–396.

Follmann, D. & Lambert, D. (1991), 'Identifiability of finite mixtures of logistics regression models.', *Journal of Statistical Planning and Inference* **27**, 375–381.

Forina, M. E. A. (1991), 'Parvus - an extendible package for data exploration, classification and correlation.', *Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno.* pp. 1–10.

Fraley, C. & Raftery, A. (1998), 'How many clusters? which clustering method? answer via model-based cluster analysis.', *Computer Journal* **41**(8), 578–588.

Fraley, C. & Raftery, A. (2002), 'Model-based clustering, discriminant analysis, and density estimation.', *Journal of American Statistical Association.* **97**(458), 611–631.

Fruhwirth-Schnatter, S. (2006), 'Finite mixture and markov switching models', *New York: Springer.* pp. 1–10.

Gamerman, D. & Lopes, H. F. (2006), '*Markov Chain Monte Carlo: Stochastic Simulation of Bayesian Inference*', *Chapman and Hall/CRC Texts in Statistics Science.* (2).

Gelman, A., Aki, V., Pasi, J., Robert, C. & Nicolas, C. (2014), 'Expectation propagation as a way of life.', *arXiv:1412.4869v1* pp. 1–29.

Gelman, A., F., Bois, Y. & Jiang, J. (1996), 'Physiological pharmacokinetic analysis using population modelling and informative prior distributions', *Journal of the American Statistical Association* **91**, 1400 – 1412.

Gelman, A. & Rubin, D. (1992a), 'Inference from iterative simulation using multiple sequences.', *Statistical Science* **7**, 457–511.

Gennari, J., Langley, P. & Fisher, D. (1989), 'Models of incremental concept formation.', *Artificial Intelligence.* **40**, 11–61.

Geoffrion, A. M. (1972), 'Generalized benders decomposition.', *Journal of Optimization Theory and Applicatios* **10**(4), 237–260.

Gershenfeld, N. (1997), 'Nonlinear inference and cluster-weighted modeling.', *Annals of the New York Academy of Sciences* **808**(1), 18–24.

Gershenfeld, N., Schoner, B. & Metois, E. (1999), 'Cluster-weighted modelling for time-series analysis.', *Physics and Media Group, MIT Media Laboratory, Cambridge, Massachusetts 02139, USA.* **1**(1), 1–10.

Geyer, C. (2011), 'Introduction to markov chain monte carlo. *Handbook of Markov Chain Monte Carlo.*', pp. 1–10.

Ghahramani, Z. & Beal, M. J. (2000), 'Variational inference for bayesian mixtures of factor analysers', *In Advances in Neural Information Processing Systems 12, Cambridge, MA. MIT Press* pp. 1–10.

Ghahramani, Z. & Jordan, M. I. (1997), 'Factorial hidden markov models', *Machine Learning.* **29**, 245–273.

Gilks, W. (1995), 'Markov chain monte carlo in practice', *Chapman and Hall/CRC.* pp. 1–20.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), 'Deep learning.', *MIT Press.* pp. 436–444.

Grün, B. & Leisch, F. (2008), 'Identifiability of finite mixtures of multinomial logit models with varying and fixed effects.', *Technical Report, Department of Statistics University of Munich* **24**, 1–17.

Graves, A. (2011), 'Practical variational inference for neural networks.', *In advances in Neural Information Processing System.* **24**, 2348–2356.

Grun, B. (2002), 'Identifizierbarkeit von multinomialen mischmodellen.', *Master's thesis, Technische, University at Wien. Kurt Hornik and Friedrich Leisch, advisors.* pp. 1–10.

Gutierrez, R., Carroll, R., Wang, N., Lee, G. & Taylor, B. (1995), 'Analysis of tomato root initiation using a normal mixture distribution.', *Biometrics* **51**(4), 1461–1468.

Hasselblad, V. (1966), 'Estimation of parameters for mixture of normal distributions.', *Technometrics* **8**, 431–444.

Hasselblad, V. (1969), 'Estimation of finite mixtures of distributions from the exponential familty.', *Journal of the American Statistical Association* **64**, 1459–1471.

Hastie, J. & Tibshirani (2004), 'Discriminant analysis by gaussian mixtures.', *Journal of Royal Statisical Society* **58**(1), 155–176.

Hastings, W. K. (1970), 'Monte carlo sampling methods using markov chains and their applications.', *Biometrika.* **57**(1), 97–109.

Hastings, W. K. (1987), 'Monte carlo is fundamentally unsound.', *Statistician, Special Issue: Practical Bayesian Statistics* **36**(2/3), 247–249.

Heckerman, D. (1996), 'A tutorial on learning with bayesian networks', *Technical Report MSR-TR-95-06, Microsoft Research* pp. 1–10.

Hennig, C. (2000), 'Identifiability of models for clusterwise linear regression.', *Journal of classification* **17**(1), 237–296.

Hestenes, M. R. (1969*a*), 'Multiplier and gradient methods', *Journal of Optimization Theory and Applications* **4**, 302–320.

Hestenes, M. R. (1969*b*), 'Multiplier and gradient methods', *In Computing Methods in Optimization Problems* pp. 1–10.

Hinton, G. E. & Camp, D. V. (1993), 'Keeping neural networks simple by minimizing the description length of the weights.', *In Sixth ACM Conference on Computational Learning Theory, Santa Cruz* pp. 1–10.

Hinton, G. E. & Roweis, S. T. (2002), 'Stochastic neighbor embedding.', *In Advances in Neural Information Processing Systems* **15**(2/3), 833–840.

Hoerl, A. E. & Kennard, R. W. (1970), 'Ridge regression: Based estimation for nonorthogonal problems', *Technometrics* **12**, 55–67.

Horton, P. & Nakai, K. (1996), 'A probablistic classification system for predicting the cellular localization sites of proteins', *Intelligent Systems in Molecular Biology, St. Louis, USA* pp. 109–115.

Hosmer, D. (1973*a*), 'A comparison of iterative mmaximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample', *Biometrics* **29**, 761–770.

Hosmer, D. (1973*b*), 'On mle of the parameters of a mixture of two normal distributions when the sample size is small.', *Computational Statistics* **1**, 217–227.

Hurvich, C. & Tsai, C. (1989), 'Regression and time series model selection in small small samples,', *Biometrika.* **76**(2), 297–307.

Hwa, K. L., Wai, K. & Philip, L. (2014), 'Zero-inflated poisson regression mixture model', *Computational Statistics and Data Analysis.* **71**, 151–158.

Ingrassia, S., Minotti, S. & Punzo, A. (2014), 'Model-based clustering via linear cluster-weighted models.', *Computational Statistics and Data Analysis,* **71**(1), 159–182.

Ingrassia, S., Punzo, A., Vittadini, G. & Minotti, S. C. (2012), 'Local statistical modeling via the cluster-weighted approach with eliptical distributions.', *Journal of Classification METRON* **29**(3), 363–401.

Ingrassia, S., Punzo, A., Vittadini, G. & Minotti, S. C. (2015), 'The generalized linear mixed cluster-weighted model.', *Journal of Classification* **32**(1), 85 – 113.

Jaakkola, T. & Jordan, M. (2000), 'Bayesian parameter estimation via variational methods.', *Statistics and Computing.* **10**, 25–37.

Jaakkola, T. & Jordan, M. I. (1996), 'Computing upper and lower bounds on likelihoods in intractable networks', *In Uncertainty in Artificial Intelligence.* .

Jaakkola, T. & Jordan, M. I. (1997), 'A variational approach to bayesian logistic regression models and their extensions', *In Artificial Intelligence and Statistics* .

Jaakkola, T. S. (2001), 'Tutorial on variational approximation methods. in m. opper and d. saad (eds.), advances in mean field methods', *MIT Press.* p. 129–159.

Jan, J. (2005), *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods*, Signal Processing and Communications, 1 edn, CRC Press.

Jeffrey, S. (1932), 'An alternative to the rejection of observations.', *Proceedings of the Royal Society of London A* **137**, 78–87.

Jensen, F. V. (1996), 'Introduction to bayesian networks', *Springer-Verlag, New York* pp. 1–10.

Jerrum, M., Sinclair, A. & Vigoda, E. (2001), 'A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries', *In ACM Symposium on Theory of Computing.* p. 712–721.

John, P., Blei, D. & Jordan, M. (2011), 'Variational bayesian inference with stochastic search.', pp. 1–8.

Jordan, M. I. (1999), 'Learning in graphical models', *MIT Press, Cambridge, MA* pp. 1–10.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. (1999), 'An introduction to variational methods for graphical models', *Machine Learning* **37**(2), 183–233.

Jorgensen, M. A. & Hunt, L. A. (1996), 'Mixture modelling clustering of data sets with categorical and continuous variables.', *In ISIS: Information, Statistics and induction in Science, D.L. Dowe, K.B. Korb, and J.J. Oliver (Eds.). Singapore: World Scientific Publishing,* pp. 375–384.

Jose, M. H. & Ryan, A. P. (2015), 'Probabilistic backpropagation for scalable learning of bayesian neural networks. *arXiv:1502.05336v2 [stat.ML]*', pp. 1–15.

Jylanki, P., Mummenmaa, A. & Vehtari, A. (2014), 'Expectation propagation for neural networks with sparsity-promoting priors', *Journal of Machine Learning* **15**(1), 1849–1901.

Jylanki, P., Vanhatalo, J. & Vehtari, A. (2011), 'Robust gaussian process regression with a student-t likelihood.', *Journal of Machine Learning Research* **12**, 3227–3257.

Kapur, J. (1989), 'Maximum entropy methods in science and engineering.', *Wiley* pp. 1–10.

Kim, A. S. I. & Wand, M. P. (2016), 'The explicit form of expectation propagation for a simple statistical model', *Electronic Journal of Statistics* **10**, 550–581.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983), 'Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm.', *Optimization by Simulated Annealing.* **220**(4598), 671–680.

Koehler, A. & Murphee, E. (1988), 'A comparison of the akaike and schwarz criteria for selecting model order.', *Applied Statistics* **37**, 187–195.

Koehler, A. & Murphee, E. (1993), 'Comparing approaches for testing the number of components in a finite mixture model.', *Computation Statistics* **9**, 65–78.

Kucukelbir, A., Ranganath, R., Gelman, A. & Blei, D. (2015), 'Automatic variational inference in stan', *In Neural Information Processing Systems* .

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. & Blei, D. M. (2016), 'Automatic differentiation variational inference', *arXiv preprint arXiv:1603.00788* .

Kullback, S. (1959), 'Information theory and statistics.', *John Wiley and Sons* pp. 1–10.

Kullback, S. & Leibler, R. (1951), 'On informationand sufficiency.', *Annals of Mathematical Statistics* **1**(22), 79–86.

Kushner, H. & Yin, G. (1997), 'Stochastic approximation algorithms and applications', *Springer New York* .

Lambert, D. (1992), 'Zero-inflated poisson regression, with an application to defects in manufacturing.', *Technometrics* **34**, 1–14.

Lasdon, L. S. (1970), 'Optimization theory for large systems.', *MacMillan* pp. 1–30.

Lauritzen, S. (1992), 'Propagation of probabilities, means and variances in mixed graphical association models.', *J American Statistical Association.* pp. 1098–1108.

Lauritzen, S. L. & Spiegelhalter, D. J. (1988), 'Local computations with probabilities on graphical structures and their application to expert systems', *Journal of the Royal Statistical Society, Series B (Methodological)* **50**(2), 157–224.

Lawrence, C. J. & Krzanowski, W. J. (1996), 'Mixture separation for mixed-mode data.', *Statistics and Computing* **6**, 85–92.

Lawrence, C. & Krzanowski, W. (1999), 'Fitting a mixture model to three-mode three-way data with categorical and continuous variables.', *Journal of Classification* **16**, 283–296.

LeCun, Y., Bengio, Y. & Hinton, G. (2015), 'Deep learning', *Nature.* **521**(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998), 'Gradient-based learning applied to document recognition.', *Proceedings of the IEEE* **86**(11), 2278–2324.

Leroux (1992a), 'Consistent estimation of a mxing distribution.', *Annals of Statistics.* **20**, 1350–1360.

Lindsay, B. G. (1995), 'Mixture models: Theory, geometry, and applications', *The Institute for Mathematical Statistics, Hayward, California,* .

Lindstrom, M. & Bates, D. (1991), 'Newton-raphson and em algorithms for linear mixed-effects models for repeated-measures data.', *Journal of the American Statistical Association.* **83**, 1014–1022.

Lo, K., Brinkman, R. & Gottardo, R. (2008), 'Automated gating of flow cytometry data via robust model-based clustering.', *Cytometry Part A* **73**(4), 321–332.

Luenberger, D. G. (1973), 'Introduction to linear and nonlinear programming.', *Addison-Wesley: Reading, MA* pp. 1–10.

Maaten, L. V. & Hinton, G. E. (2008), 'Visualizing data using t-stochastic neighbor embedding.', *Journal of Machine Learning Research* **9**, 2579–2605.

MacKay, D. J. C. (1997), 'Ensemble learning for hidden markov models', *Technical report, Cavendish Laboratory, University of Cambridge,* pp. 1–10.

Magnus, R., Stegle, O., Sharp, K. & Winn, J. (2009), 'Inference algorithms and learning theory for bayesian sparse factor analysis', *International Workshop on Statistical-Mechanical Informatics, Journal of Physics: Conference Series* **197**, 1–11.

Maitra, R. (2009), 'Initializing partition-optimization algorithms.', *Transactions on Computational Biology and Bioinformatics.* **6**, 144–157.

Maria, G., Eleftherios, F., Eirini, F. & George, L. (2016), 'Characterization of "y-eye: a low-cost benchtop mouse-sized gamma camera for dynamic and static imaging studies"', *World Molecular Imaging Society* pp. 398–407.

Matthias, G. & Bangti, J. (2014), 'Expectation propagation for nonlinear inverse problems with an application to electrical impedance tomography. *Journal of Computational Physics*', (513 - 535).

Maybeck, P. (1982), 'Stochastic models, estimation and control, chapter 12.7.'.

Mazza, A., Punzo, A. & Ingrassia, S. (2018), 'flexcwm: A flexible framework for cluster-weighted models. journal of statistical.', *Journal of Statistical Software* **86**(2), 1–30.

McLachlan, G. & Basford, K. (1988), 'Mixture models: Inference and applications to clustering.', *Marcel Dekker, New York.* pp. 1–10.

McLachlan, G. & Peel, D. (2000), 'Finite mixture models.', *Wiley Series In Probability and Statistics Applied Probability and Statistics Section,Wiley, New York.* **1**(1), 1–438.

McNicholas, P. (2010), 'Model-based classification using latent gaussian mixture models.', *Journal of Statistical Planning and Inference.* **140**(5), 1175–1181.

McNicholas, P. & Murphy, T. (2008), 'Parsimonious gaussian mixture models.', *Statistics and Computing.* **18**(3), 285–296.

McQuarrie, A., Shumway, R. & Tsai, C. (1997), 'The model selection criterion aicu,', *Statistics and Probability Letters.* **34**(3), 285–292.

Meijer, E. & Ypma, J. Y. (2008), 'A simple identification proof for a mixture of two univariate normal distributions.', *Journal of Classification* pp. 1–10.

Metropolis, N., Rosenbluth, A. W., Teller, M. N. & Teller, E. (1953), 'Equation of state calculations by fast computing machines.', *Journal of Chemical Physics* **21**, 1087–1092.

Miele, A., Cragg, E. E., Iver, R. R. & Levy, A. V. (1971), 'Use of the augmented penalty function in mathematical programming problems, part 1', *Journal of Optimization Theory and Applications* **8**, 115–130.

Miele, A., Cragg, E. E. & Levy, A. V. (1971), 'Use of the augmented penalty function in mathematical programming problems, part 2', *Journal of Optimization Theory and Applications* **8**, 131–153.

Miele, A., Mosely, P. E., Levy, A. V. & Coggins, G. M. (1972), 'On the method of multipliers for mathematical programming problems', *Journal of Optimization Theory and Applications* **10**, 1–33.

Minka, T. (2001), Expectation propagation for approximate bayesian inference., *in* '*In proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, ed.* J. Breese and D Koller', Vol. 20, XML, pp. 362–369.

Minka, T. (2004), 'Power expectation propagation.', *Technical report, Microsoft Research, Cambridge.* pp. 1–10.

Minka, T. (2005), 'Divergence measures and message passing. *Technical report, Microsoft Research, Cambridge.*', **1**, 1–7.

Muthen, B. & Asparouhov, T. (2009), 'Multilevel regression mxiture analysis.', *J. R. Stat. Soc. Ser. A (Stat. Soc.)* **3**(172), 639–657.

Nair, V. & Hinton, G. E. (2010), 'Rectified linear units improve restricted boltzmann machines', *In Proceedings of the 27th International Conference on Machine Learning* **27**, 807–814.

Neal, R. M. (1992), 'Connectionist learning of belief networks.', *Artificial Intelligence* **56**, 71–113.

Neal, R. M. & Hinton, G. E. (1999), 'A new view of the em algorithm that justifies incremental and other variants.', *In M. I. Jordan (Ed.), Learning in Graphical Models* pp. 355–368.

Newcomb, S. (1886), 'A generalized theory of the combination of observeations so as to obtain the best result..', *American Jounral of Mathematics* **8**, 343–366.

Opper, M. & Winther, O. (1999), 'Bayesian approach to on-line learning.', *On-line Learning in Neural Networks. Cambridge University Press* pp. 1–10.

Parisi, G. (1988), 'Statistical field theory', *Addison-Wesley* pp. 1–10.

Pearl, J. (1988), 'Probabilistic reasoning in intelligent systems: Networks of plausible inference', *Morgan Kaufmann Publishers, San Francisco, CA* pp. 1–10.

Pearson, K. (1894), 'Contributions to the theory of mathematical evolution.', *Philosophical Transactions of the Royal Society of London* **185**, 71–110.

Peters, B. & Coberly, W. (1976), 'The numerical evaluation of the maximum likelihood estimate of mixture proportions.', *Communications in Statistics - Theory and Methods* **5**, 1127–1135.

Peterson, C. & Anderson, J. R. (1987), 'A mean field theory learning algorithm for neural networks', *Complex Systems* **1**, 995–1019.

Priebe, C. E. (1994), 'Adaptive mixtures.', *Journal of the American Statistical Association* (89), 796–806.

Pritchard, M. & Wilson, S. (2003), 'Using emotional and social factors to predict student success', *College Student Development,* **44**(1), 18–28.

Pronzato, L., Walter, E., Venot, A. & Lebruchec, J, F. (1984), 'A general-purpose global optimizer: Implimentation and applications.', *Mathematics and Computers in Simulation.* **26**(5), 412–422.

Punzo, A. (2014), 'Flexible mixture modeling with the polynomial gaussian cluster-weighted model.', *arXiv:1207.0939v1* **1**(1), 1–25.

R, D. T. (2019), 'R: A language and environment for statistical computing.', *R Foundation for Statistical Computing, Vienna, Autria.* .

Rand, W. (1971), 'Objective criteria for the evaluation clustering methods,', *Journal of the American Statistical Association.* **66**(336), 846–850.

Rényi, A. (1961), 'On measures of entropy and information.', *Fourth Berkeley symposium on mathematical statistics and probability* **1**, 1–10.

Rao, C. R. (1948), 'The utility of multiple measurements in problems of biological classification.', *Journal of the Royal Statistical Society B* **10**, 159–203.

Robbins, H. & Monro, S. (1951), 'A stochastic approximation method.', *Annals of Mathematical Statistics.* **22**, 400–407.

Robert, C. (1996), 'Mixtures of distributions: inference and estimation.', *In Markov Chain Carlo in Practice, W.E. Gilks, S. Richardson, and D.J. Spiegelhalter, London: Chapman and Hall.* pp. 441–464.

Robert, C. & Casella, G. (2004), 'Monte carlo statistical methods', *Springer Texts in Statistics. Springer-Verlag, New York, NY.* .

Roberts, S., Husmeier, D., Rezek, I. & Penny, W. (1998), 'Bayesian approaches to gaussian mixture models', *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **20**(11), 1133–1142.

Rockafellar, R. T. (1970), 'Convex analysis', *Princeton University Press* pp. 1–20.

Roeder, K. & Wasserman, L. (1997), 'Practical density estimation using mixtures of normals.', *Journal of the American Statistical Association* **92**, 894–902.

Sam, W. (1995), 'Donor of database.', *Department of Computer Science, University of Tasmania, GPO Box 252C, Hobart, Tasmania 7001, Australia* p. 1.

Saul, L. K., Jaakkola, T. & Jordan, M. I. (1996), 'Mean field theory for sigmoid belief networks', *Journal of artificial intelligence research* **4**, 61–76.

Schlattmann, P. (2009), 'Medical applications of finite mixture models.', *Springer-Verlag* pp. 1–10.

Schmidhuber, J. (2015), 'Deep learning in neural networks: An overview', *Neural Networks.* **16**, 85–117.

Schwarz, G. (1978), 'Estimating the dimension of a model.', *Annals of Statistics* **6**, 461–464.

Schwarz, H. R. (1988), 'Finite element methods.', *Academic Press.* pp. 1–10.

Scrucca, L., Fop, M., Murphy, T. B. & Raftery, A. E. (2016), 'mclust: clustering, classification and density estimation using gaussian finite mixture models', *The R Journal* **8**(1), 205–233.

Seeger, M. (2008), 'Bayesian inference and optimal design for the sparse linear model.', *Journal of Machine Learning Research* (9), 759–813.

Shor, N. Z. (1985), 'Minimization methods for non-differentiable functions', *SpringerVerlag* pp. 1–20.

Subedi, S., Punzo, A., Ingrassia, S. & McNicholas, P. (2012), 'Clustering and classification via cluster-weighted factor analyzer', *arXiv:1209.6463v1 [stat.ME]* **1**(1), 1–36.

Swets, J., Dawes, R. & Monahan, J. (2000), 'Better decisions through science,', *Academic Press, New York.* **283**, 82–87.

Teicher, H. (1961), 'Identifiability of mixtures', *Annals of Mathematical Statistics.* **34**, 244–248.

Teicher, H. (1963), 'Identifiability of finite mixtures.', *The Annals of Mathematical Statistics* **34**, 1265–1269.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *J. R Stat. Society* **58**, 267–288.

Titterington, D., Smith, A. & Markov, U. (1985), 'Statistical analysis of finite mixture distributions.', *New York: John Wiley and Sons* pp. 1–10.

Ueda, N., Nakano, R., Gharamani, Z. & Hinton, G. (2000), 'Smem algorithm for mixture models.', *Neural Computation* **12**(9), 2109–2128.

Wang, P., Cockburn, I. & Putterman, M. (1998), 'Analysis of patent data: a mixed poisson regression model approach.', *journal Business Econometric Statistics* **16**(1), 27–41.

Wang, P., Putterman, M., Cockburn, I. & Le, N. (1996), 'Mixed poisson regression models with covariate dependent rates.', *Biometrics* **52**(2), 381–400.

Warwick, J. N., Tracy, L. S., Simon, R. T., Andrew, J. C. & Wes, B. F. (1994), 'The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait", sea fisheries division, technical', *Sea Fisheries Division, Technical* **No. 48**, ISSN 1034–3288.

Waterhouse, S., MacKay, D. J. C. & Robinson, T. (1996), 'Bayesian methods for mixtures of experts.', *In Advances in Neural Information Processing Systems 8, Cambridge,* pp. 1–10.

Wedel, M. (2002), 'Concomitant variables in finite mixture mixture models.', *Statistica Neerlandica* **56**(3), 362–375.

Wedel, M., DeSarbo, W., Bult, J. & Ramaswamy, V. (1993), 'A latent class poisson regression model for heterogeneous count data.', *Journal of Application Econometrics* **8**, 397–411.

Wedel, M. & Kamakura, W. (2001), 'Market segmentation: Conceptual and methodological foundations (2nd ed.).', *Boston MA: Kluver Academic Publisher* pp. 1–10.

Weldon, W. F. R. (1892), 'Certain correlated variations in carcinus vulgaris.', *Proceedings of the Royal Society of London* **51**, 2–21.

Weldon, W. F. R. (1893), 'On certain correlated variations in carcinus moenas.', *Proceedings of the Royal Society of London* (54), 318–329.

Whittaker, J. (1990), 'Graphical models in applied multivariate analysis.', *New York: Wiley.* **55**, 1–17.

Wolfe, J. (1965), 'A computer program for the computation of maximum likelihood analysis of types.', *Research Memo. SRM San Diego: U.S. Naval Personnel Research Activity.* **65**, 12.

Wolfe, J. (1967), 'Normix: Computational methods for estimating the parameters of multivariate normal mixtures of distributions.', *Research Memo. SRM San Diego: U.S. Naval Personnel Research Activity.* **68**, 2.

Wolfe, J. (1970), 'Pattern clustering by multivariate mixture analysis.', *Multivariate Behavior Research.* **5**, 329–350.

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

Yingzhen, L., Jose, M. & Richard, E. (2018), 'Stochastic expectation propagation. *arXiv:1506.04132v2[Stat.ML].*', pp. 1–10.

Zoeter, O. & Heskes, T. (2005), 'Gaussian quadrature based expectation propagation', *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* pp. 445–452. The Society for Artificial Intelligence and Statistics.