UNIVERSITY OF CAMBRIDGE

Generalised Bayesian Matrix Factorisation Models

Shakir Mohamed

St John's College
University of Cambridge

THESIS

Submitted for the degree of
Doctor of Philosophy, University of Cambridge

FEBRUARY 2011

I, SHAKIR MOHAMED, confirm that *this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text*. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

I also confirm that this thesis is below 60,000 words and contains less than 150 figures, in fulfilment of the requirements set by the degree committee for the Department of Engineering at the University of Cambridge.

Abstract

Factor analysis and related models for probabilistic matrix factorisation are of central importance to the unsupervised analysis of data, with a colourful history more than a century long. Probabilistic models for matrix factorisation allow us to explore the underlying structure in data, and have relevance in a vast number of application areas including collaborative filtering, source separation, missing data imputation, gene expression analysis, information retrieval, computational finance and computer vision, amongst others. This thesis develops generalisations of matrix factorisation models that advance our understanding and enhance the applicability of this important class of models.

The generalisation of models for matrix factorisation focuses on three concerns: widening the applicability of latent variable models to the diverse types of data that are currently available; considering alternative structural forms in the underlying representations that are inferred; and including higher order data structures into the matrix factorisation framework. These three issues reflect the reality of modern data analysis and we develop new models that allow for a principled exploration and use of data in these settings. We place emphasis on Bayesian approaches to learning and the advantages that come with the Bayesian methodology. Our port of departure is a generalisation of latent variable models to members of the exponential family of distributions. This generalisation allows for the analysis of data that may be real-valued, binary, counts, non-negative or a heterogeneous set of these data types. The model unifies various existing models and constructs for unsupervised settings, the complementary framework to the generalised linear models in regression.

Moving to structural considerations, we develop Bayesian methods for learning sparse latent representations. We define ideas of weakly and strongly sparse vectors and investigate the classes of prior distributions that give rise to these forms of sparsity, namely the scale-mixture of Gaussians and the spike-and-slab distribution. Based on these sparsity favouring priors, we develop and compare methods for sparse matrix factorisation and present the first comparison of these sparse learning approaches. As a second structural consideration, we develop models with the ability to generate correlated binary vectors. Moment-matching is used to allow binary data with specified correlation to be generated, based on dichotomisation of the Gaussian distribution. We then develop a novel and simple method for binary PCA based on Gaussian dichotomisation. The third generalisation considers the extension of matrix factorisation models to multi-dimensional arrays of data that are increasingly prevalent. We develop the first Bayesian model for non-negative tensor factorisation and explore the relationship between this model and the previously described models for matrix factorisation.

Acknowledgements

My time at the University of Cambridge has been filled with a wealth of memorable experiences and I have been privileged to be surrounded by a multitude of truly inspirational people. Foremost amongst these people has been my supervisor, Prof. Zoubin Ghahramani. Zoubin has been a friend and teacher, confidant and ally, and a supervisor of the highest calibre. I will continue to strive towards the standards of excellence that he has laid. I would also like to thank Dr Carl Rasmussen who has always been a valuable source of perspective. Dr Katherine Heller has been an extraordinary mentor, collaborator and source of motivation - thank you. I must also thank Prof. Tshilidzi Marwala for encouraging me to pursue a PhD, and one at the University of Cambridge, no less.

I wish to thank all the members of the lab who have endured my hours of verbiage and who made the lab such a wonderful place in which to work. Thank you to: Finale Doshi-Velez, Frederik Eaton, David Knowles, Simon Lacoste-Julien, Peter Orbanz, Pedro Ortega, Yunus Saatchi, Mikkel Schmidt, Jurgen van Gael, Sinead Williamson, and so many others who I'm sure I have left out.

Much is due to my friends who have kept me smiling. I thank: Ruth Mokgokong, Grace Wong, Desha Osborne, Jared Rossouw, Lindelwa Dalamba and Andrew MacDonald. Marc Deisenroth has been, and continues to be a pillar of strength, my best friend and greatest supporter – much appreciation goes to him.

Most importantly, I thank my family, especially my parents, Hoosain and Naeema, and my brothers, Shaheed and Naeem, for their support, love and encouragement over the last three years. Without them none of this would have been possible.

I am grateful to *St John's College* for the graduate access studentship scheme as well as for 10th term funding, and for the wealth of experiences and opportunities I have been exposed to. I sincerely thank the *Commonwealth Scholarship and Fellowship Programme* (CSFP) for awarding me with a Commonwealth scholarship to the UK, which made my coming to Cambridge a reality. The modern Commonwealth is truly enhanced by this programme and the interaction, development and friendship that the programme promotes. I also wish to acknowledge the role of the *Cambridge Commonwealth Trust* (CCT) and the *British Council* (BC) in the administration of my award. The BC in particular ensured that my time in the UK was as smooth as possible. I also thank the *National Research Foundation* (NRF), South Africa, for support and the invaluable role they play in supporting South African researchers. Travel funds were obtained with the generous support of the PASCAL network of excellence, the NIPS foundation and the International Society for Bayesian Analysis.

Contents

List of Figures

List of Tables

List of Examples

Notes on Notation

Symbol	Description
$\mathcal{X}$	Generally, a P-way tensor of dimensions $M_1 \times \ldots \times M_P$.
$\mathbf{X}$	A matrix, usually considered to have dimensions $D \times N$.
$\mathbf{x}_n$	A column vector, being the nth column of the matrix $\mathbf{X}$.
x	Scalar variable.
x_{ij}	Element i,j of the matrix $\mathbf{X}$.
$\Omega \backslash \Omega_j$	All elements of the set Ω excluding the jth item.
$\mathbb{I}(x > 0)$	The indicator function that $x > 0$.
$\mathbb{E}_{p(x)}[x]$	Expectation of random variable x under the distribution $p(x)$.
$x_i \perp\!\!\!\perp x_j \vert \theta$	Conditional independence between x_i and x_j given θ.

Distribution	
Uniform	$\mathcal{U}(x \in \mathbb{R} \vert [a,b]) = \frac{1}{b-a}$ for $a < x < b$
Gaussian	$\mathcal{N}(\mathbf{x} \in \mathbb{R}^D \vert \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} \vert\boldsymbol{\Sigma}\vert^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$
Gamma	$\mathcal{G}(x \in \mathbb{R}^+ \vert \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$
Bernoulli	$\mathcal{B}(x \in \{0,1\} \vert \pi) = \pi^x (1-\pi)^{(1-x)}$
Beta	$\beta(\pi \in [0,1] \vert \alpha, \gamma) = \frac{\Gamma(\alpha+\gamma)}{\Gamma(\alpha)\Gamma(\gamma)} \pi^{(\alpha-1)} (1-\pi)^{(\gamma-1)}$
Poisson	$\mathcal{P}(x \in \mathbb{N} \vert \lambda) = \frac{1}{x!} \exp(-\lambda)\lambda^x$
Exponential	$\mathcal{E}(x \in \mathbb{R}^+ \vert \lambda) = \lambda \exp(-\lambda x)$
Laplace	$\mathcal{L}(x \in \mathbb{R} \vert \lambda) = \frac{1}{2}\lambda \exp(-\lambda \vert x \vert)$
Canonical Exponential Family	$\text{Expon}(\mathbf{x} \vert \boldsymbol{\eta}) = h(\mathbf{x}) \exp\left(S(\mathbf{x})^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\right)$
Conjugate Exponential Family	$\text{Conj}(\boldsymbol{\eta} \vert \boldsymbol{\lambda}, \nu) = \exp\left(\boldsymbol{\lambda}^\top \boldsymbol{\eta} - \nu A(\boldsymbol{\eta}) - f(\boldsymbol{\lambda}, \nu)\right)$

Chapter 1

Latent Variable Models and Probabilistic Inference

Matrix factorisation models are some of the most popular models in current statistical practice. The name stems from the intuition in the use of this broad class of models: to learn the set of factors that underlie data and the phenomena from which the data was obtained. Statistically, these underlying factors are represented by *latent* or *hidden variables*, variables whose realisations are not observed directly but must rather be inferred given other *manifest variables*, whose realisations are observed. Models with latent variables provide a rich tool-kit with which to explore many problems: studying the underlying behaviour in biological systems, surveying the themes embedded in document archives, unpacking customer shopping behaviours or removing noise in experimental data; and are indispensable in the specification of generative descriptions of data. As a result, we will often refer to matrix factorisation models as latent variable models. In this thesis, we will develop new latent variable models that advance our current understanding and usage of this important class of models. This introductory chapter motivates the general use of latent variables in the analysis of real data along with the statistical tools that will be used for inference. The primary objective here is to enframe the development of latent variable models in later chapters and to highlight the place and importance of the work to be developed in the general study of statistical models with latent variables.

1.1 The Ubiquitous Latent Variable

Models with latent variables hold a central role in in the analysis of data in a diverse set of research areas spanning machine learning, statistics, economics, psychology, computational biology, geography and political science. The omnipresence of latent variables is now widely recognised, though this may be obscured in some settings

where latent variables exist under a variety of alternative names, including random effects, common factors and latent traits or classes. What distinguishes latent variables from model parameters is that for every observation $\mathbf{x}_n \in \mathbb{R}^D$, there is a corresponding latent variable $\mathbf{v}_n \in \mathbb{R}^K$, for the nth observation. Therefore, the number of latent variables grows with the size of the data, whereas the number of parameters in a model is usually fixed irrespective of the data size. The latent variables provide an underlying representation of the data and low dimensional representations are obtained if the dimensionality of the latent variables is less than that of the observed data, $K < D$. If $K > D$ then the latent representation is referred to as over-complete.

The latent variable models considered in this thesis encompass at least four broad motivations for the use of latent variables:

Finding 'true' values. In many data analysis problems, we assume that a true value for noisy measurements exists and consider the statistical problem of finding this true value. A data point x_n for the nth observation is generated as:

$$x_n = v_n + \epsilon_n,$$

where v_n is a latent variable that is the true value of the signal and ϵ_n is the measurement error. It is this motivation that is often used for the popular method of principal components analysis (PCA) (Joliffe, 2002). In PCA, the true measurement is assumed to be a low dimensional embedding that lies in a subspace. PCA is an important foundational model of concern in this thesis, and will be described in more detail in chapter 2.

Hypothetical explanations of data. Latent variables can be considered to represent hypothetical factors underlying each of the observed data points. Here, we consider the observed data to be indirect indicators of meaningful latent factors, such as factors of 'self-esteem' or 'positive preference' in psychological studies, or 'political impact' in political science. This is important since it is the motivation for the use of many statistical models with latent variables and their use in exploratory data analysis. Chapters 2 and 3 look at applications of this type.

Learning flexible distributions. Latent variables can be used to generate multivariate distributions with a particular dependence structure. One such situation is the analysis of count based data with an excessive number of zero-entries. Such a situation can arise in the analysis of manufacturing defects. In a properly calibrated system, there are no defects in the product manufactured – a defect-free mode. In a miscalibrated system, the number of defects is subject to random fluctuations – a defect-prone mode. Modelling in this setting uses a zero-inflated Poisson distribution (Lambert, 1992) to account for the higher

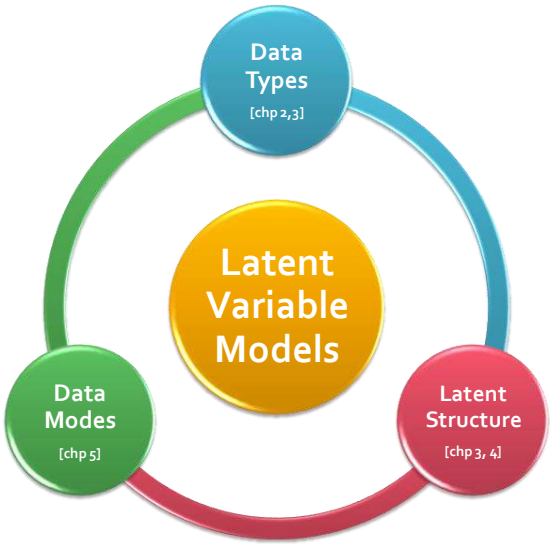


Figure 1.1: Diagrammatic thesis outline showing the focus of each of the chapters in the context of their contribution to modelling with latent variables.

number of zero defects, since the system is most often in the defect-free mode. A latent variable is introduced to indicate membership to the defect-free or defect-prone mode, and allows a flexible distribution to be learnt, a task which cannot be achieved using the standard Poisson distribution. The use of latent variables for learning flexible distributions akin to this situation is discussed in chapter 3.

Studying thresholding effects. Latent variables are also useful for the analysis of coarsened variables. It is not uncommon for a continuous variables v_n to be dichotomised or thresholded, resulting in an indicator response such that $x_n = \mathbb{I}(v_n > 0)$, where the latent variable v_n is seen as the propensity to be in the category indicated by $x_n = 1$. The use of latent variables in exactly this manner will be discussed in chapter 4.

Newer types of data are generated each day from a wide range of technologies such as high throughput genome sequencing, blog entries and posts using social networking media, customer purchasing decisions at supermarkets involving detailed purchasing histories, new measurement systems in hospitals and manufacturing facilities, or traffic patterns in a city. With this new data comes the need for an advancement of available models, a need for increasing accuracy and sophistication to provide competitive advantage, and a need to understand the complex phenomena that underlie these modern systems. This thesis is motivated by these needs, to advance latent variable models to consider: newer kinds of data types that are prevalent, newer kinds of structure underlying the observed data and newer data structures in which the data may be stored. We will expand upon this triad of concerns, figure 1.1, in each of the ensuing chapters, where new models will be developed for these analysis

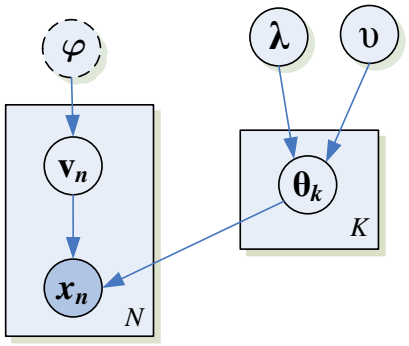


Figure 1.2: Graphical model showing the form of a general latent variable model.

Latent Variable	Model
Gaussian	Factor analysis/PCA
Multinomial	Gaussian mixture model
Dirichlet	Partial Membership model
Laplace	Sparse latent feature model

Table 1.1: Models obtained using the generic latent variable model structure under differing distributional assumptions for the latent variables.

problems. We will provide a succinct and intuitive understanding of the models and learning tools used and will describe how these new models fit into the wider context of modelling with latent variables.

1.2 Models for Matrix Factorisation

We will use the term latent variable model to refer to any model that can be described by the generic graphical model of figure 1.2. The plate notation represents replication of variables. For this class of models, the observed data $\mathbf{X}$ consists of N observations $\mathbf{x}_n$, which are D-dimensional vectors. The observed data $\mathbf{X}$ is assumed to be factorised into a set of latent factors $\mathbf{V}$ and factor loadings $\boldsymbol{\Theta}$. The set of latent factors or underlying representation is given by the latent variables $\mathbf{v}_n \in \mathbb{R}^K$, with $K < D$ and the set of all factors is the matrix $\mathbf{V}$. The parameters $\boldsymbol{\theta}_k$ are referred to as factor loadings and the set of loadings is represented by the matrix $\boldsymbol{\Theta}$. We will describe specific distributional assumptions for random variables in the graphical model in the subsequent chapters.

Importantly, latent variable models of this form are models for *matrix factorisation*, since the likelihood $p(\mathbf{X}|\mathbf{V}, \boldsymbol{\Theta})$ depends only on the product of the matrix factors $\boldsymbol{\Pi} = \boldsymbol{\Theta}\mathbf{V}$. This is conceptually simple while being a very flexible modelling approach for use in a wide range of tasks.

Consider a Gaussian noise model of the form: $\mathbf{x}_n = \boldsymbol{\Theta}\mathbf{v}_n + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$. Given this specification, the choice of prior distribution for the latent variables $\mathbf{v}_n$ in figure 1.2, spans a broad class of models in popular use. Table 1.1 lists various distributional assumptions for the latent variables and the corresponding model that is then obtained, showing the generality of the latent variable model construction. Generalised latent variable modelling is thus the study of various aspects of this generality. For the remainder of this chapter, we will describe the principles used in

constructing models for probabilistic matrix factorisation and latent variable models. We summarise the important properties of the exponential family of distributions, and review aspects of Bayesian inference and posterior computation using Markov chain Monte Carlo methods. We conclude this chapter with an outline of the remaining chapters and the key contributions made in each.

1.3 The Exponential Family of Distributions

The exponential family is an important family of distributions that emphasises the shared properties of many standard distributions, including the Binomial, Poisson, Gamma, Beta, Multinomial and the Gaussian distributions. The exponential family of distributions allows for a singular discussion of the inferential properties associated with members of the family, and forms the basis of an important class of models known as generalised linear models. In this section we will review the aspects of the exponential family of distributions relevant for our discussion.

1.3.1 The one-parameter exponential family

A one-parameter exponential family is a parameterised family of density functions that can be written in the following form:

$$p(x|\theta) = h(x) \exp\left(\eta(\theta)S(x) - B(\theta)\right), \tag{1.1}$$

for $\mathbf{x} \in \mathbb{R}^d$ and real-valued functions h, η, B, which are not unique. The space of parameters $\theta \in \Theta$ for which $B(\theta)$ is defined is referred to as the mean parameter space. It is often more useful to index the model by η rather than θ, giving rise to the *canonical one-parameter exponential family*:

$$q(x|\eta) = h(x) \exp\left(\eta S(x) - A(\eta)\right) \tag{1.2}$$

$$A(\eta) = \ln \int h(x) \exp(\eta S(x)) dx, \tag{1.3}$$

where η is refereed to as the *natural parameter* of the distribution and $S(x)$ as the *sufficient statistics*. $A(\eta)$ is the *log-partition* or cumulant function, which must be finite and ensures that $q(x)$ is normalised. $h(x)$ is not of particular interest, but reflects the underlying measure with respect to which $q(x|\eta)$ is a density. The function $\eta(\theta)$ is referred to as the link function, since it is a function from the mean parameters to the natural parameters. The space Ω, which contains all η such that $A(\eta)$ is finite, is referred to as the *natural parameter space*. The set Ω is a convex set with the functions $A(\eta)$ being convex functions and is of importance for inference with such distributions (as will be discussed in section 2.2.3). Table 1.2 provides a useful listing of the exponential family forms for several well-known distributions.

Table 1.2: Several well known exponential families listing their log-partition functions $A(\eta)$, conjugate dual functions $A^*(\theta)$, corresponding Bregman divergence $B(x\|\theta)$ and canonical link functions $\eta(\theta)$.

Family	$A(\eta)$	$A^*(\theta)$	$B_{A^*}(x\|\theta)$	$\eta(\theta)$
Bernoulli	$-\log(1+\exp(\eta))$	$\theta\ln\theta$–$(1$–$\theta)\ln(1$–$\theta)$	$\ln(1+\exp(x^*\theta))$ $x^*=2x-1$	$\ln\left(\frac{\theta}{1-\theta}\right)$
Exponential	$-\log(-\eta)$	$\theta\ln\theta-\theta$	$x\ln\left(\frac{x}{\theta}\right)-(x-\theta)$	θ
Poisson	$\exp(\eta)$	$-(1+\ln\theta)$	$\frac{x}{\theta}-\ln\left(\frac{x}{\theta}\right)-1$	$\ln(\theta)$
Multinomial	$\ln\left(1+\sum_{i=1}^{k-1}\exp(\eta_i)\right)$	$\sum_{j=1}^{k}\theta_j\ln\left(\frac{\theta_j}{N}\right)$	$\sum_{j=1}^{k}x_j\ln\left(\frac{x_j}{\theta_j}\right)$	$\ln\left(\frac{\theta_j}{1-\sum_{i=1}^{k-1}\theta_i}\right)$
Gaussian (location family)	$\frac{1}{2\sigma^2}\eta^2$	$\frac{1}{2\sigma^2}\theta^2$	$\frac{(x-\theta)^2}{2\sigma^2}$	θ

Example 1.1: ***The Bernoulli Family***

The Bernoulli distribution is a one-parameter exponential family, which can seen by rewriting the density function.

$$p(x|\theta) \quad = \quad \theta^x(1-\theta)^{1-x} = \exp\left\{\log\left(\frac{\theta}{1-\theta}\right)x + \log(1-\theta)\right\}, \qquad (1.4)$$

where the approach taken is to rewrite the density as the exponential of the logarithm of the original distribution and rearranging to obtain the required form. This is an exponential family distribution employing the logit link function:

$$k=1, \quad \eta(\theta)=\log\left(\frac{\theta}{1-\theta}\right), \quad S(x)=x, \quad h(x)=1.$$
$$B(\theta)=-\log(1-\theta), \quad \text{or } A(\eta)=-\log(1+\exp(\eta)) \text{ (in canonical form).} \quad \square$$

1.3.2 The k-parameter exponential family

One-parameter exponential families are naturally indexed by a one-dimensional real parameter η. Common one parameter distributions are the Bernoulli and the Poisson. Other distributions admit k-parameter exponential families, which is the parameterised collection of density functions of the form:

$$p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x})\exp\left(\sum_{j=1}^{k}\eta_j(\boldsymbol{\theta})S_j(\mathbf{x}) - B(\boldsymbol{\theta})\right), \qquad (1.5)$$

for observed data $\mathbf{x}\in\mathbb{R}^d$, natural parameter vector $\boldsymbol{\eta}(\boldsymbol{\theta})=[\eta_1,\ldots,\eta_k]^\top$, and sufficient statistics $S(\mathbf{x})=[S_1(\mathbf{x}),\ldots,S_k(\mathbf{x})]^\top$, with $S_1,\ldots,S_k$ on $\mathbb{R}^d$. This can be expressed in the canonical form as:

$$q(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})\exp\left(S(\mathbf{x})^\top\boldsymbol{\eta} - A(\boldsymbol{\eta})\right). \qquad (1.6)$$

We will use the shorthand $\mathrm{Expon}\,(\mathbf{x}|\boldsymbol{\eta})$ to refer to the exponential family of distributions given by equation (1.6).

Example 1.2: ***The Gaussian Family***

The standard Gaussian distribution $p(x|\boldsymbol{\theta}) = \mathcal{N}(x|\mu, \sigma^2)$, may be rewritten as:

$$p(x|\boldsymbol{\theta}) \quad = \quad \exp\left\{\frac{\mu}{\sigma^2}x - \frac{x^2}{2\sigma^2} - \frac{1}{2}\left(\frac{\mu^2}{\sigma^2} + \ln(2\pi\sigma^2)\right)\right\}. \tag{1.7}$$

This corresponds to a two-parameter exponential family with:

$$\theta_1 = \mu, \theta_2 = \sigma^2$$

$$\eta_1(\boldsymbol{\theta}) = \frac{\mu}{\sigma^2}, \quad \eta_2(\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}$$

$$S_1(x) = x, \quad S_2(x) = x^2$$

$$B(\boldsymbol{\theta}) = -\frac{1}{2}\left(\frac{\mu^2}{\sigma^2} + \ln(2\pi\sigma^2)\right), \quad A(\boldsymbol{\eta}) = -\frac{1}{2}\left(\frac{\eta_1^2}{2\eta_2} + \ln\left(\frac{\pi}{\eta_2}\right)\right).$$

□

1.3.3 Conjugate Families of Prior Distributions

Distributions that are members of the exponential family also have natural conjugate prior distributions. By conjugate we mean that for a given probability distribution $p(x|\boldsymbol{\theta})$, we seek a prior $p(\boldsymbol{\theta})$ such that the posterior distribution has the same functional form as the prior. For a k-parameter exponential family distribution $p(x|\boldsymbol{\theta})$, the conjugate distribution on $\boldsymbol{\theta}$ is given by the $(k+1)$-parameter exponential family:

$$p(\boldsymbol{\theta}) = \exp\left(\sum_{j=1}^{k}\eta_j(\boldsymbol{\theta})\lambda_j - \lambda_{k+1}B(\boldsymbol{\theta}) - f(\boldsymbol{\lambda})\right).$$

This is an exponential family with sufficient statistics given by $\{\eta_j(\boldsymbol{\theta}), -B(\boldsymbol{\theta})\}$ and natural parameters $\boldsymbol{\lambda}$. It will be convenient to use the canonical form, representing the $(k+1)$th parameter as ν for clarity:

$$q(\boldsymbol{\lambda}, \nu) = \exp\left(\sum_{j=1}^{k}\eta_j\lambda_j - \nu A(\boldsymbol{\eta}) - f(\boldsymbol{\lambda}, \nu)\right). \tag{1.8}$$

We will use the shorthand $\mathrm{Conj}\,(\boldsymbol{\lambda}, \nu)$ for the conjugate exponential family given by equation (1.8).

Example 1.3: ***The Beta Prior***

The Beta distribution is the conjugate distribution to the Bernoulli distribution described in example 1.1. The density function can be written as:

$$\begin{aligned} p(\theta|a,b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{(a-1)}(1-\theta)^{(b-1)} \\ \ln p(\theta|a,b) &= \ln\left(\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\right) + (a-1)\ln\theta - (b-1)\ln\left(\frac{1}{1-\theta}\right), \end{aligned} \quad (1.9)$$

which corresponds to the $(k+1)$-parameter exponential family with:

$$(\lambda,\nu) = \{(a-1),(b-1)\}, \quad \eta(\theta) = \ln\theta, \quad f(\lambda,\nu) = \ln\left(\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\right)$$

$$B(\theta) = -\log(1-\theta), \quad \text{or } A(\eta) = -\log(1+\exp(\eta)) \text{ (in canonical form).}$$

□

Example 1.4: ***The Conjugate Beta-Bernoulli Model***

Consider the simple conjugate Beta-Bernoulli model:

$$\mathbf{z}_n \sim \prod_k \mathcal{B}(z_{nk}|\pi_k) = \prod_k \pi_k^{z_{nk}}(1-\pi_k)^{(1-z_{nk})} \quad (1.10)$$

$$\pi_k \sim \beta(\pi_k|\alpha,\gamma) = \frac{\Gamma(\alpha+\gamma)}{\Gamma(\alpha)\Gamma(\gamma)}\pi_k^{(\alpha-1)}(1-\pi_k)^{(\gamma-1)}, \quad (1.11)$$

where $\mathcal{B}(z_{nk}|\pi_k)$ is the Bernoulli distribution with probability π_k and $\beta(\pi_k|\alpha,\gamma)$ is the Beta distribution with shape α and scale γ. The $\mathbf{z}_n$ are independent given $\boldsymbol{\pi}$, with $n = 1,\ldots,N$. Due to conjugacy, the posterior distribution for π_k is a Beta distribution and is:

$$p(\pi_k|\boldsymbol{z}) = \beta(\pi_k|\bar{\alpha},\bar{\gamma}) \quad (1.12)$$

$$\bar{\alpha} = \alpha + \sum_n z_{nk} \qquad \bar{\gamma} = \gamma + N - \sum_n z_{nk}. \quad (1.13)$$

□

1.3.4 Exponential Families and Bregman Divergences

The Bregaman divergence is a generalised distance measure that is closely related to any discussion of Exponential family distributions, since it can be shown that their exists a unique Bregman divergence corresponding to every regular exponential family (Bregman, 1967; Azoury and Warmuth, 2001; Banerjee et al., 2005). The Bregman

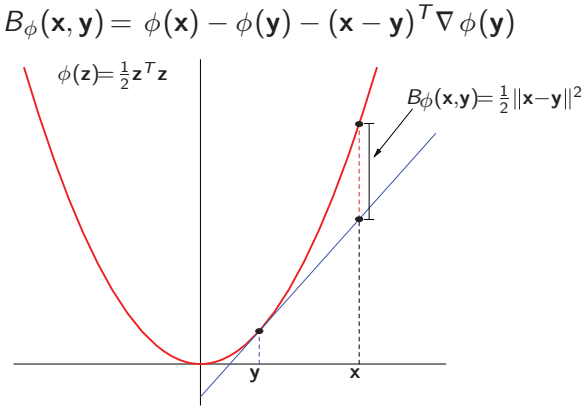


Figure 1.3: Illustration of the Bregman distance between points x and y, with a quadratic function ϕ, which corresponds to the Euclidean distance.

divergence between x, y, for a differentiable and strictly convex function ϕ, is:

$$B_\phi(x,y) \quad = \quad \phi(x) - \phi(y) - \nabla\phi(y)(x-y), \tag{1.14}$$

where $\nabla\phi(y)$ represents the gradient of ϕ evaluated at y. Intuitively, the Bregman divergence measures the convexity of the function ϕ. The divergence measures the increase in $\phi(x)$ over $\phi(y)$ above linear growth given by the slope $\nabla\phi(y)$. This is shown diagrammatically in figure 1.3 considering a quadratic function, in which case the Bregman divergence is equivalent to the Euclidean distance. In general, every Bregman divergence is non-negative and is equal to zero if and only if its two arguments are equal. Popular distance measures such as the Euclidean distance, logistic loss, Itakura-Saito distance and the KL-divergence can be expressed in this form and are Bregman divergences.

The relationship between the Bregman divergence and the exponential family can be seen by examining the negative-log probability of an exponential family distribution, written as:

$$\ln p(x|\theta) \quad = \quad -B_{A^*}(x,\theta(\eta)) + \ln b_{A^*}(x), \tag{1.15}$$

which is the sum of a Bregman divergence and a function that is constant with respect to θ and can therefore be ignored. $\theta(\eta)$ is the inverse canonical link function that transforms natural parameters η to their corresponding mean parameters θ. The properties of the convex function ϕ are well studied and for the exponential family of distributions, ϕ is the conjugate dual of the log-partition function $A^*(\theta)$, giving $\phi(\theta) = A^*(\theta)$. The conjugate dual function corresponds to the negative entropy of the particular exponential family distribution (Wainwright and Jordan, 2006, thm 3.4).

The use of Bregman divergences thus provides an alternative view of learning with exponential family distributions: learning with generalised loss functions given

by the Bregman divergence. In addition, the divergence has convex properties that are useful in designing learning algorithms for models involving these distributions. Table 1.2 lists some well known exponential family distributions with their corresponding Bregman divergences.

Example 1.5: ***The Bernoulli Family (cont.)***

The Bernoulli family was first discussed in example 1.1. The canonical link and inverse link functions were derived as:

$$\eta(\theta) = \ln\left(\frac{\theta}{1-\theta}\right), \quad \theta(\eta) = \frac{1}{1+\exp(-\eta)}.$$

The conjugate function A^*, which is the negative entropy of the Bernoulli distribution is given by:

$$A^*(\theta) = \theta\ln(\theta) + (1-\theta)\ln(1-\theta).$$

Using this convex function, the Bregman divergence using equation (1.14) with $x^* = 2x - 1$ is thus:

$$\begin{aligned} B_{A^*}(x\|\theta) =& A^*(x) - A^*(\theta) - (x-\theta)\nabla A^*(\theta) \\ =& x\ln\tfrac{x}{\theta} + (1-x)\ln\left(\tfrac{1-x}{1-\theta}\right) \end{aligned} \tag{1.16}$$

□

1.4 Probabilistic Modelling and Bayesian Inference

Throughout the thesis, we develop probabilistic approaches for matrix factorisation. A probabilistic approach provides a principled approach to learning and a means of dealing with uncertainties involved in the data generation and model specification processes. A probabilistic model is specified by providing the joint-probability distribution of all variables used in characterising the learning problem. Since complex settings are usually considered, latent variables are introduced to aid the modelling process. Often the generation of data is thought of as a sequence of realisations from a hierarchy of random variables, such as figure 1.2. The model of interest is then the joint distribution of any latent variables v, model parameters θ, and the observed data x: $p(x, v, \theta)$. The task is then to learn the values of these unknown parameters and latent variables from the observed data.

The *likelihood* function is of key importance in probabilistic modelling, and is the probability that a model with any particular parameter setting assigns to the

observed data. The likelihood is thought of as a function of the parameters θ and encapsulates the ability of the chosen parameters to explain the given data: $L(\theta|x_1, \ldots, x_n) = p(x_1, \ldots, x_n|\theta)$. In *maximum likelihood inference*, a point estimate of parameters is determined that maximises this likelihood given the observed data, though in practice the log-likelihood is maximised instead. Approaches to learning rely on the theory of optimisation, which is immense and allows for effective and scalable algorithms for learning to be designed.

Using *Bayesian inference*, rather than finding point estimates, we can instead learn the posterior distribution of parameters. Bayesian statistics is the powerful branch of statistics with which we can determine the posterior distribution of parameters conditioned on the observed data by using Bayes' theorem:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}, \tag{1.17}$$

where $p(\theta)$ is called the *prior* probability distribution and embodies the prior belief of plausible parameter values in various regions of the parameter space. The introduction of the prior allows the likelihood to be transformed into a proper distribution over parameters. The Bayesian approach allows for the natural inclusion of prior knowledge and provides a mechanism for dealing with uncertainty by learning posterior distributions rather than point estimates of the parameters. Bayesian inference has many other advantages, such as a built-in regularisation and safeguards against model overfitting. We will discuss the advantages of Bayesian methods in further detail in the next chapter.

The following integration problems are central to Bayesian statistics:

Normalisation. To obtain the posterior distribution, the normalising factor in Bayes' theorem (1.17) must be computed:

$$p(x) = \int p(x|\theta)p(\theta)d\theta.$$

Marginalisation. Marginal distributions may often be of interest, particularly when latent variables are involved:

$$p(\theta|x) = \int p(\theta, v|x)dv.$$

Expectation. Often summary statistics are sought, of the form:

$$\mathbb{E}_{p(x|\theta)}\left[f(x)\right] = \int f(x)p(x|\theta)dx,$$

for some function $f(x)$, e.g. if the mean is sought, then $f(x) = x$.

These integration problems are typically intractable and must be approximated by some means: Markov chain Monte Carlo methods are popular in this regard and will be described further in the next section.

We will exemplify the principles of probabilistic inference that have been highlighted here in the ensuing chapters, showing more precisely the specification of the joint probabilities and the description of the generative processes of data. We will develop both maximum likelihood and Bayesian inference techniques in this thesis, but will place a strong emphasis on Bayesian approaches to learning. A more thorough and complete discussion of probabilistic modelling and inference can be found in the books by Bishop (2006); MacKay (2003); Bernardo and Smith (1994) and are reference works on many fundamental aspects of probabilistic modelling that will be referenced throughout this thesis.

1.5 Markov Chain Monte Carlo Methods

Markov chain Monte Carlo (MCMC) methods are established tools for solving the typically intractable integration problems central to Bayesian statistics that were just described. MCMC methods trace their history to the work of Metropolis and Ulam (1949) and the subsequent generalisation of this work to the Metropolis-Hastings method (Hastings, 1970). But the lack of computational resources in earlier research curtailed the wider use of MCMC as a method for inference. With modern computing technology, MCMC is now widespread throughout statistical practice, with this popularity being attributed to the work of Gelfand and Smith (1990). The work of Neal (1993) is also highly significant, especially in popularising MCMC in the machine learning community.

The merits of MCMC as an approach for inference in comparison to other inference methods such as variational methods or expectation propagation are not discussed, though this is of relevance and interest. MCMC is a wide area of research and the texts by Gilks et al. (1995); Robert and Casella (2004); MacKay (2003); Bishop (2006) provide excellent discussions covering the breadth of current MCMC practice. In addition, the review papers by Neal (1993) and Andrieu et al. (2003) are highly informative. We make use of three well established MCMC methods in this thesis: Gibbs sampling, Hybrid Monte Carlo sampling and slice sampling. We provide algorithmic descriptions of these methods and defer technical aspects of these methods to the reference texts provided.

1.5.1 Gibbs Sampling

Gibbs sampling is arguably the most widely applied of MCMC methods. The Gibbs sampler was given its name by Geman and Geman (1984) for problems in image restoration and subsequently popularised by Gelfand and Smith (1990). Gibbs sampling aims to generate samples from the posterior distribution of $\boldsymbol{\theta}$ that is partitioned into a vector of components $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$. Although it may be hard to sample from the joint-posterior, it is assumed that it is easy to simulate from the *full conditional distributions* $p(\theta_i|\{\theta_j, j \neq i\})$. Implementing the Gibbs sampler begins with initial guesses for the θ_i denoted $\theta_1^{(0)}, \ldots, \theta_d^{(0)}$. Sampling then iterates through the following steps, for iteration t:

$$\theta_1^{(t)} \sim p(\theta_1|\{\theta_j^{(t-1)}, j \neq 1\}), \tag{1.18}$$

$$\theta_2^{(t)} \sim p(\theta_2|\theta_1^{(t)}, \{\theta_j^{(t-1)}, j > 2\}), \tag{1.19}$$

$$\vdots$$

$$\theta_i^{(t)} \sim p(\theta_i|\{\theta_k^{(t)}, k < i\}, \{\theta_j^{(t-1)}, j > i\}), \tag{1.20}$$

$$\vdots$$

$$\theta_d^{(t)} \sim p(\theta_d|\{\theta_j^{(t)}, j \neq d\}). \tag{1.21}$$

As t approaches infinity, the joint distribution can be shown to approach the joint distribution of $\boldsymbol{\theta}$. In order for Gibbs sampling to produce samples from the correct distribution, the resulting Markov chain must be ergodic. This implies that none of the conditional distributions should be zero anywhere. If this is satisfied, then any point in the space can be reached from any other point using updates of each of the component variables. For t^* sufficiently large, the set of samples $\boldsymbol{\theta}^{(t^*)}$ can be regarded as one simulated draw from the posterior distribution. L such samples can be generated and used to compute any required posterior moments.

Samples obtained from Gibbs iterations are always accepted, making Gibbs sampling simply the repeated simulation from full conditional distributions. Many models make use of conjugate pairs of distributions, which allow the required full conditional distributions to reduce analytically to closed-form distributions, for which efficient sampling methods exist. If only two iterating steps are needed, then Gibbs sampling is often referred to as data augmentation. If any of the full conditional distributions are not amenable to sampling from a known closed-form distribution, then samples must be simulated using any other sampling technique – the default choice being the Metropolis-Hastings method. This sampling scheme is then referred to as *Metropolis-within-Gibbs* sampling. Since sampling from non-conjugate distributions is more involved, many models use conjugate pairs of distributions so that inference can be performed using Gibbs sampling. If at each

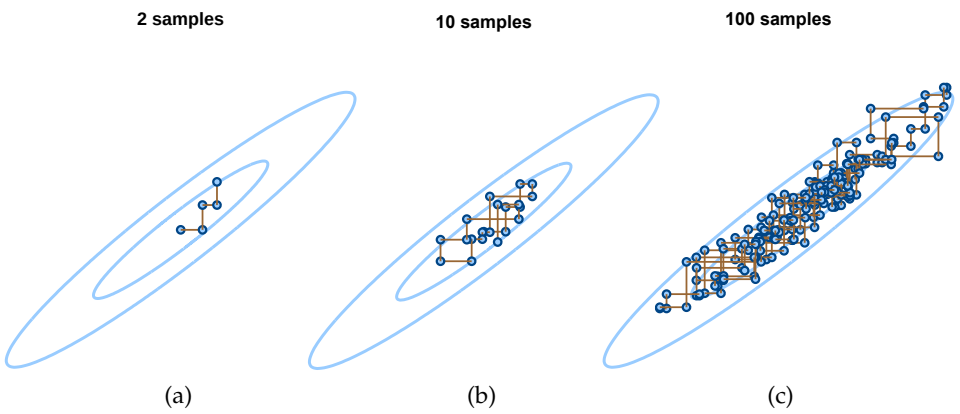


Figure 1.4: Sampling from the two-dimensional Gaussian distribution showing the progression of Gibbs sampling.

stage, the maximum of the conditional distribution is used instead of samples being drawn, then this method is referred to as Iterated Conditional Modes (ICM) (Kittler and Föglein, 1984), making ICM a greedy approximation to Gibbs sampling.

There is typically strong positive correlation between the values of $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t+1)}$. If independent samples are required, then thinning of the samples must be applied, where samples at $t, t+s, t+2s$ and so on are used for spacing s. Figure 1.4 shows the behaviour of Gibbs sampling when simulating from a two-dimensional Gaussian distribution. The exploration of the space is effective, though the correlation between the samples results in slower mixing (as seen by the lack of samples in the lower left corner after 100 samples).

1.5.2 Hybrid Monte Carlo Sampling

The Hybrid Monte Carlo sampling approach is the first of two auxiliary variable sampling methods we discuss in this chapter. In the auxiliary variable sampling framework, instead of sampling from the distribution $p(\boldsymbol{\theta})$, samples are obtained from an augmented distribution $p(\boldsymbol{\theta}, \mathbf{u})$, where $\mathbf{u}$ is a set of auxiliary variables. The idea of sampling with auxiliary variables originated in physics with the work of Swendsen and Wang (1987) and is central to Hybrid Monte Carlo sampling (HMC) discussed here and in slice sampling discussed in the next section.

Hybrid Monte Carlo was first described by Duane et al. (1987) and is based on the simulation of Hamiltonian dynamics. The details of the physical underpinnings describing Hamiltonian dynamics and its appropriateness for MCMC are best described in the work of Neal (1993, 2010). Consider generating samples from the distribution $p(\boldsymbol{\theta}|\boldsymbol{\psi})$, with $\boldsymbol{\psi}$ being any relevant hyperparameters; an auxiliary variable $\mathbf{u}$ will be used. Intuitively, HMC combines auxiliary variables with gradient

information from the joint-probability space to improve mixing of the Markov chain. The gradient acts as a force that results in more effective exploration of the sample space. HMC can be used for sampling from continuous distributions for which the density function can be evaluated (up to a known constant). This makes HMC particularly amenable to sampling in non-conjugate settings where full conditional distributions cannot be obtained, but for which joint-probability densities can be computed. The derivatives of the log-density function must also exist.

For HMC, a Potential energy function and a Kinetic energy function is defined, whose sum forms the Hamiltonian energy:

$$\mathcal{H}(\boldsymbol{\theta}, \mathbf{u}) = \mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\psi}) + \mathcal{K}(\mathbf{u}) \qquad \text{(Hamiltonian Energy)} \tag{1.22}$$

$$\mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\psi}) = -\ln p(\boldsymbol{\theta}|\boldsymbol{\psi}) \qquad \text{(Potential Energy)} \tag{1.23}$$

$$\mathcal{K}(\mathbf{u}) = -\tfrac{1}{2}\mathbf{u}^\top \mathbf{u} \qquad \text{(Kinetic Energy)} \tag{1.24}$$

The Hamiltonian can be seen as the log of the augmented distribution to be sampled from: $p(\boldsymbol{\theta}, \mathbf{u}|\boldsymbol{\psi}) = p(\boldsymbol{\theta}|\boldsymbol{\psi})\mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{I})$. The gradient of the Potential energy is defined as: $\Delta(\boldsymbol{\theta}) = \frac{\partial \mathcal{E}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$. Each iteration of HMC has two steps. In the first step, we assume that an initial sample (state) for $\boldsymbol{\theta}$ is given and generate a Gaussian variable $\mathbf{u}$ for the momentum (line 4, algorithm 1.1). In the second step, we simulate Hamiltonian dynamics, which follows the equations of motion to move the current sample and momentum to a new state. The Hamiltonian dynamics must be discretised for implementation and the most popular discretisation is known as the leapfrog method (lines 7-11, algorithm 1.1). The leapfrog approximation is simulated for τ steps using a step-size ϵ. The values of $\boldsymbol{\theta}$ and $\mathbf{u}$ at the end of the leapfrog steps form the proposed state, which is accepted using the metropolis criterion (line 15, algorithm 1.1):

$$\min\left(1, \exp(-\mathcal{H}(\boldsymbol{\theta}^*, \mathbf{u}^*) + \mathcal{H}(\boldsymbol{\theta}, \mathbf{u}))\right). \tag{1.25}$$

Marginal samples from $p(\boldsymbol{\theta})$ are obtained by ignoring $\mathbf{u}$. The full set of steps needed for HMC are described by algorithm 1.1.

HMC requires the selection of two free parameters. The number of leapfrog steps τ, and the step-size ϵ. In general the step-size should be chosen to ensure that the sampler's rejection rate is between 25% - 35%. It is also preferable to have a large number of leapfrog steps since this reduces the random walk behaviour of the sampling (Neal, 1993). Typically, 50 leapfrog steps are used in the applications in this thesis. The selection of these parameters can be challenging, but there exists a great deal of guidance in choosing these parameters and in tuning HMC for optimal performance in general. The review chapter by Neal (2010) provides a wealth of guidance in tuning HMC and many other aspects of its application in practical situations. Theoretical analysis also exists regarding optimal tuning, the most recent

Algorithm 1.1: Hybrid Monte Carlo (HMC) Sampling

Evaluate Gradient $\mathbf{g} = \boldsymbol{\Delta}(\boldsymbol{\theta})$ with initial $\boldsymbol{\theta}$ `// g = gradE(theta)`
Evaluate Energy $E = \mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\psi})$ `// E = findE(theta)`
for L *iterations* **do**
  Initialise new momentum $\mathbf{u}$ drawn from a Gaussian
  Calculate: $\mathcal{K}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^\top\mathbf{u}$ and $\mathcal{H} = \mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\psi}) + \mathcal{K}(\mathbf{u})$
  $\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}$; $\quad \boldsymbol{g}^{new} \leftarrow \boldsymbol{g}$;
  for τ *leapfrog steps* **do**
    $\mathbf{u} \leftarrow \mathbf{u} - \frac{\epsilon}{2}\boldsymbol{g}$ `// Make half-step in` u
    $\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}^{new} + \epsilon\mathbf{u}$ `// Make a step in theta`
    $\boldsymbol{g}^{new} \leftarrow \boldsymbol{\Delta}(\boldsymbol{\theta}^{new})$ `// gradE(thetaNew)`
    $\mathbf{u} \leftarrow \mathbf{u} - \frac{\epsilon}{2}\boldsymbol{g}^{new}$ `// make half step in` u
  $E^{new} = \mathcal{E}(\boldsymbol{\theta}^{new}|\boldsymbol{\psi})$ `// Enew = findE(thetaNew)`
  Calculate $\mathcal{K}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^\top\mathbf{u}$
  Hamiltonian $\mathcal{H}^{new} \leftarrow E^{new} + \mathcal{K}(\mathbf{u})$
  if `rand()` $< \exp(-(\mathcal{H}^{new} - \mathcal{H}))$ **then**
    Accept $\leftarrow$ `True`
    $\boldsymbol{g} \leftarrow \boldsymbol{g}^{new}$; $\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{new}$; $\quad E \leftarrow E^{new}$
  else
    Accept $\leftarrow$ `False`

of which is described by Beskos et al. (2010). Figure 1.5 shows the behaviour of HMC in sampling from the two-dimensional Gaussian. Qualitatively comparing this to figure 1.4, HMC is much more effective than Gibbs sampling in exploring the space.

1.5.2.1 Hybrid Monte Carlo with Constrained Variables

Many modelling problems involve the use of random variables that may be constrained, e.g. be non-negative or bound between [0,1]. Hybrid Monte Carlo is still amenable in this setting, but requires an adjustment to the energy function that is used. The method to be described here will be referred to as the *transformation method*. Consider the Bayesian modelling of data $\mathbf{X}$ with constrained parameters c and prior distribution $p(c)$. The posterior distribution to be sampled from is:

$$p(c|\mathbf{X}) \propto p(\mathbf{X}|c)p(c). \tag{1.26}$$

To perform Hybrid Monte Carlo sampling in this setting, the constrained variables c must first be transformed to unconstrained variables u, using any suitable transformation: $c = T(u)$. The determinant of the Jacobian of the change of variables must be included: $J(u) = \frac{\partial c}{\partial u} = \frac{\partial T(u)}{\partial u}$, giving the new posterior probability as:

$$p(u|\mathbf{X}) \quad = \quad |J(u)|\, p(c|\mathbf{X}). \tag{1.27}$$

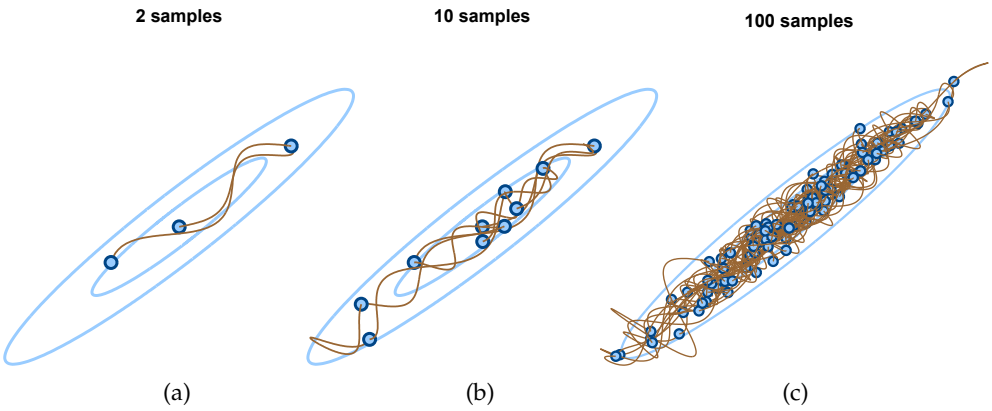


Figure 1.5: Sampling from the two-dimensional Gaussian distribution showing the progression of HMC sampling. The lines represent the simulated path followed during the leapfrog iterations.

Commonly used transformations include the exponential function for non-negatively constrained parameters, or the softmax function for parameters bound on a simplex.

The usual HMC algorithm 1.1 can be applied after transforming the constrained variables to unconstrained variables and making the following adjustments to the potential energy function and its derivatives:

$$\begin{aligned} \mathcal{E}(u) &= -\ln p(u|\mathbf{X}) = -\ln p(\mathbf{X}|c) - \ln p(c) - \ln J(u), && (1.28) \\ \frac{\partial \mathcal{E}(u)}{\partial u} &= \frac{\partial \mathcal{E}(c)}{\partial c}\frac{\partial c}{\partial u} \quad \text{(Chain Rule)} \\ &= -\frac{\partial \ln p(\mathbf{X}|c)}{\partial c}\frac{\partial c}{\partial u} - \frac{\partial \ln p(c)}{\partial c}\frac{\partial c}{\partial u} - \frac{\partial \ln J(u)}{\partial u}. && (1.29) \end{aligned}$$

It is especially important not to forget to apply the chain rule consistently to the derivatives of the potential energy function, since this can be easily overlooked.

Example 1.6: ***Sampling from the Log-Normal Distribution***

To show that the adjustments for sampling with constrained variables give the correct results in general settings, consider sampling from the random variable c with log-Normal distribution, which is bound to the range $[0, \infty)$ and has the density function:

$$p(c|0,1) = \frac{1}{c\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\ln c)^2\right), \quad c \geq 0. \tag{1.30}$$

Using the transformation: $T(u) : c = \exp(u)$, sampling in the unconstrained space involves the following energy function:

$$\mathcal{E}(u) = \mathcal{E}(c) - \ln|J(u)| = -\ln(\tfrac{1}{c}) - \frac{1}{2}\ln^2 c - \ln c = \frac{1}{2}u^2. \tag{1.31}$$

By inspection, this is the form of a Gaussian distribution $\mathcal{N}(0, 1)$. This then, verifies the well know technique of sampling from the log-Normal distribution based on transformations of a Gaussian random variable using the exponential function (Devroye, 1986). □

Neal (2010) discusses an alternative means by which to handle constraints on model parameters, based on modifying the leapfrog method used in simulating the dynamics, and will be referred to as the *splitting method*. Any constraints on subsets of the parameters can be handled, such as $c \leq b_u$, $c \geq b_l$ or both. The key aspect of this adjustment involves the specification of a Potential energy function that is infinite for any parameter values that violate the constraints, giving such parameters zero probability. Further details of this approach require more discussion of the leapfrog discretisation than has been provided here and are thus deferred to Neal (2010). We will use the transformation method in our applications of HMC since suitable transformations are known in all constrained cases that we consider.

1.5.3 Slice Sampling

Slice sampling (Neal, 1997; Damien et al., 1999; Neal, 2003) is a further example of an auxiliary variable sampler and is a generalised version of the Gibbs sampler. Like Gibbs sampling, a slice sampling chain has no rejections but is more straightforward to implement than Gibbs sampling, and can be shown to be more efficient than simple Metropolis updates (Neal, 2003). Slice sampling introduces an auxiliary variable u, known as the slice level, to construct an extended density $q(\theta, u) = 1$ if $0 \leq u \leq p(x)$ and 0 otherwise. This results in the following full conditional distributions:

$$p(u|\theta) = \mathcal{U}(u|[0, p(\theta)]), \tag{1.32}$$

$$p(\theta|u) = \mathcal{U}(\theta|\{\theta : p(\theta) \geq u\}), \tag{1.33}$$

where $\mathcal{U}(\theta|A)$ is the uniform distribution over the region A. Slice sampling thus alternates between sampling the slice level u, and then sampling θ in the interval $A = \{\theta : p(\theta) \geq u\}$. If there are multiple dimensions, slice sampling operates by cycling though each of the dimensions with all other dimensions fixed. If the region A is known, then slice sampling is easy to implement. A simple strategy for determining the region A involves growing a region (called a bracket) around the current value $\theta^{(t-1)}$ using a step-size w, and testing that $p(\theta) \geq u$. This process is continued with the bracket expanding until the condition is no longer true. More sophisticated methods for determining the bracket have also been developed and are discussed by Neal (2003); Skilling and MacKay (2003).

Slice sampling is an appealing MCMC method since all that is required for a successful implementation is the evaluation of the joint-density function (up to a

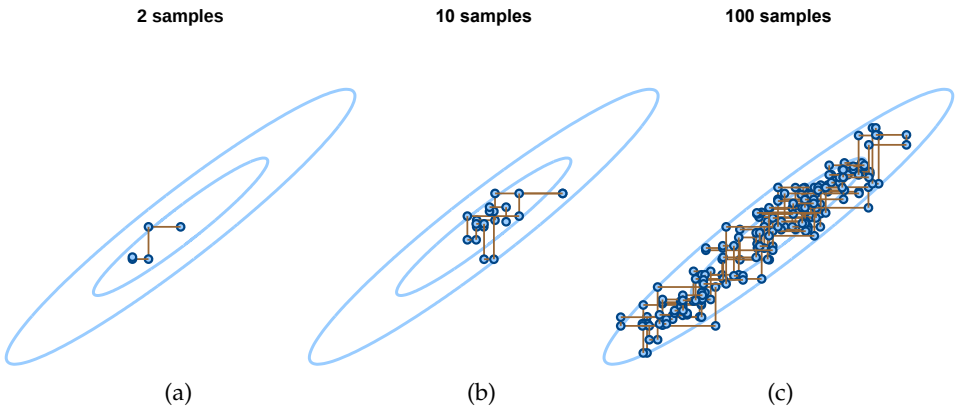


Figure 1.6: Sampling from the two-dimensional Gaussian distribution showing the progression of slice sampling.

known constant) and the specification of a step-size w. Other methods such as Gibbs sampling require the derivation of full conditional distributions, or require the specification of many free parameters needed for tuning as with Hybrid Monte Carlo. Figure 1.6 shows the slice sampling behaviour in sampling the two dimensional Gaussian. Recent advances in slice sampling allow for joint updates to be made instead of in a co-ordinate-wise fashion in situations where Gaussian latent variables are used (Murray et al., 2010), enhancing the attractiveness of slice sampling as a method for sampling.

1.5.4 Monitoring Chain Convergence

The objective of MCMC methods is to obtain samples from the target posterior distribution and to explore its characteristics. If the resulting sequence has not converged, then inferences that are obtained may not be sensible. As a result, a great deal of research is dedicated to determining when a Markov chain has mixed sufficiently and the length of the chain required to ensure suitable mixing. Most approaches focus on monitoring the convergence of the chain with the aim of rejecting the null hypothesis that the chain has *not* reached convergence. Rejecting this hypothesis does not imply that the chain has actually converged, but rather that there is no reason to suspect lack of convergence given the current test – a stronger statement cannot be made. The standard practice is to evaluate the chain convergence using at least two convergence assessment techniques. The two convergence assessment methods used here are: Gelman's potential scale reduction factor (PSRF) (Gelman et al., 2004) and the Brooks's hairiness index (Brooks and Roberts, 1998).

The potential scale reduction factor (PSRF), denoted $\hat{R}$, evaluates the convergence of scalar quantities of interest to the sampling problem, by examining the performance of multiple chains with dispersed starting points. The replication of

5 chains is usually enough, and is what is used in this thesis. The between- and within-chain variance is computed, with the intuition that at convergence these two quantities should be the same. A detailed discussion of the computation of the PSRF appears in Gelman et al. (2004). Guidance in using the PSRF is simply that the value should be low, with high values indicating that further simulation of the chain may improve inferences about the target distribution. In general, the value of $\hat{R} < 1.1$ is the oft-suggested criterion with which to decide when to stop the chain (Gelman et al., 2004, pp. 297).

The Hairiness index, denoted $\hat{H}$, is based on the CUSUM method for convergence monitoring (Robert and Casella, 2004, pp 481). CUSUM monitors how often derivatives of the sampler statistics of interest change in sign: infrequent changes in sign indicate that the sampler may not have reached convergence. The hairiness index is a measure of these changes in derivative and is usually plotted with 95% confidence intervals. Problems with convergence are flagged when the index lies outside the confidence interval. Further details regarding the computation of the hairiness index is deferred to Brooks and Roberts (1998) or Robert and Casella (2004).

While details are omitted here regarding these convergence methods, there is a great deal of research in this area. Robert and Casella (2004) provide a deeper discussion on theoretical aspects of convergence and other relevant empirical methods of convergence assessment. The review papers by Neal (1993); Cowles and Carlin (1996) and the books by Gelman et al. (2004); Gilks et al. (1995) are also very useful for wider context in this area.

1.6 Thesis Outline

In the forthcoming chapters we will advance latent variable modelling in three ways, by: expanding model scope regarding the types of data that are considered, considering alternative structure underlying the observed data, and learning with data stored in more complex data structures. Each chapter begins by describing a broad motivation for the discussion in the chapter and moves to develop a set of models and inference algorithms that improve on currently available methods. We evaluate all models using synthetic and real data, and include application studies that aim to demonstrate the practical use of the new models for exploratory analysis and system design. Each chapter also includes an 'in context' section that places the work of the chapter in historical context, describes related work, and emphasises where the contribution of the chapter fits in the wider context.

The focus and contributions of each chapter of this thesis are:

Chapter 2. We follow the natural evolutionary path for matrix factorisation models by developing a framework for latent variable models generalised to the exponential family. This establishes the complementary framework for for unsupervised learning, which exists for regression as the generalised linear models. This exponential family generalisation extends the scope of latent variable models to dichotomous, categorical, counts and non-negative data, or heterogeneous set of these data types. We clarify the relationship between many existing models using our exponential family framework. We develop a fully Bayesian model that overcomes many of the problems of maximum likelihood leaning and demonstrate a new method for dealing with factor identifiability.

Chapter 3. Building on the framework presented in chapter 2, we develop and contrast models for learning sparse latent representations. This is an important structural aspect underlying many data sets and provides valuable insight in many applications. We show how sparse unsupervised models can be constructed, what classes of priors are applicable and the dilemmas that this may pose, and develop both maximum likelihood and Bayesian learning approaches. Importantly, we present the first comparison of such methods and provide useful guidance for the implementation of sparse models.

Chapter 4. We develop a novel and simple approach for the principal components analysis of binary data based on analysing dichotomised or thresholded underlying Gaussian variables. Using this approach, we gain an understanding of the effects of the dichotomisation process and methods for the analysis of large, sparse binary data. We demonstrate an efficient algorithm for matching moments between a correlated binary distribution and a latent Gaussian distribution. Our algorithm allows for sampling of correlated binary variables with desired means and covariance, gives insight into the implications of dichotomising a Gaussian distribution, and by combining Gaussian dichotomisation with efficient methods for computing principal components, demonstrates a simple method for the principal components analysis of binary data.

Chapter 5. We develop a novel Bayesian model for data expressed as tensors or multi-dimensional arrays of data. This type of data is generated increasingly often, especially in factorial experiments that consider outcomes under varying conditions. We employ latent variables to learn representations of each of the tensor modes. We focus on the popular class of non-negative factorisations and discuss the applications of this model class. The relationship between the new non-negative Bayesian tensor factorisation and the the matrix factorisation models considered in the previous chapters, is described along with an account of related approaches to the probabilistic modelling of tensors.

Chapter 6. This concluding chapter summarises the contributions of this thesis, explores its emergent themes and examines the scope for future work.

Chapter 2

Generalising Latent Variable Models to the Exponential Family

We begin the exposition of this chapter with the important statistical framework for linear Gaussian models. This framework, coupled with an understanding of the shared properties of members of the exponential family of distributions, allows for the construction of a class of unsupervised linear latent variable models generalised to the exponential family. We portray the historical development of such an exponential family generalisation, describe the important properties of the construction, develop a method for fully Bayesian learning and demonstrate the efficacy of the new class of generalised latent variable models through empirical analysis.

2.1 Linear Gaussian Models

Linear Gaussian models form an important statistical framework that employs Gaussian latent variables and assumes Gaussian noise (Roweis and Ghahramani, 1999). Many well known models such as principal components analysis (PCA) (Pearson, 1901; Hotelling, 1933; Joliffe, 2002), factor analysis (FA) (Spearman, 1904; Bartholomew and Knott, 1999), Gaussian mixture models (Newcomb, 1886; Titterington et al., 1985) and hidden Markov models fall within this framework. Of particular interest to this chapter is the subclass of static linear Gaussian models, which allow latent representations of i.i.d. data to be inferred and to which both principal components analysis and factor analysis belong.

Principal components analysis exemplifies the form of latent variable model that we consider here. Since the initial ideas for PCA were established by Pearson (1901), PCA has become one of the most popular methods for linear latent variable modelling. PCA is a method for dimensionality reduction that searches for a mapping

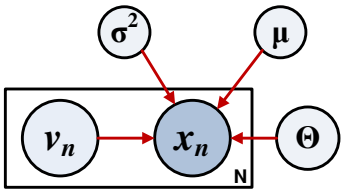


Figure 2.1: Graphical model for probabilistic PCA. The plate notation represents replication of variables and the shaded node represents the observed data.

from observed data $\mathbf{x} \in \mathbb{R}^D$ to a lower dimensional representation $\mathbf{v} \in \mathbb{R}^K$ with $K < D$; the mapping between the two is given by the eigenvectors corresponding to the K-largest eigenvalues of the data covariance matrix. Due to this conceptual simplicity, PCA is now a much relied upon tool for dimensionality reduction, feature extraction, data visualisation and image and signal processing.

A probabilistic interpretation of PCA that falls into the framework of linear Gaussian models can be given (Tipping and Bishop, 1997; Roweis, 1998), and is described using the probabilistic graphical model of figure 2.1. The graphical model describes the generative process whereby an observed data point $\mathbf{x}_n$ is considered to be a noise-corrupted version of the true data $\widetilde{\mathbf{x}}_n$ that lies in a subspace, under the assumption of Gaussian noise. A latent variable $\mathbf{v}_n$ is introduced for each observed data point and represents the principal component subspace. The model assumes a Gaussian prior for each of the latent variables $\mathbf{v}_n$, as well as a Gaussian conditional distribution $p(\mathbf{x}_n|\mathbf{v}_n)$:

$$p(\mathbf{v}_n) = \mathcal{N}(\mathbf{v}_n|\mathbf{0}, \mathbf{I}), \tag{2.1}$$

$$p(\mathbf{x}_n|\mathbf{v}_n) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\Theta}\mathbf{v}_n + \boldsymbol{\mu}, \sigma^2\mathbf{I}). \tag{2.2}$$

The $D \times K$ matrix $\boldsymbol{\Theta}$ represents the K principal components and σ^2 is the scalar variance of the conditional distribution. The negative log-likelihood yields an objective function that is equivalent to the usual PCA objective function, which minimises the Euclidean distance between the data and its reconstruction. Following this specification, all marginal and conditional distributions are Gaussian. A fully Bayesian specification includes a Gaussian prior on the matrix $\boldsymbol{\Theta}$, as a set of K independent D-dimensional Gaussian distributions:

$$p(\boldsymbol{\Theta}|\boldsymbol{\lambda}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\theta}_k|\mathbf{0}, \lambda_k\mathbf{I}). \tag{2.3}$$

Given this specification, the log-joint probability for probabilistic PCA, ignoring all constant terms is:

$$\begin{aligned} \ln p(\mathbf{X}, \mathbf{V}, \boldsymbol{\Theta}) &= \ln p(\mathbf{X}|\mathbf{V}, \boldsymbol{\Theta}) + \ln p(\mathbf{V}) + \ln p(\boldsymbol{\Theta}|\boldsymbol{\lambda}) \\ &= -\frac{1}{2}\left(\sum_n \left(\frac{1}{\sigma^2}(\mathbf{x}_n - \boldsymbol{\Theta}\mathbf{v}_n)^\top(\mathbf{x}_n - \boldsymbol{\Theta}\mathbf{v}_n) + \mathbf{v}_n{}^\top\mathbf{v}_n\right) - \sum_k \frac{1}{\boldsymbol{\lambda_k}}\boldsymbol{\theta}_k{}^\top\boldsymbol{\theta}_k\right). \end{aligned} \tag{2.4}$$

This probabilistic specification of PCA has a number of advantages: probabilistic PCA specifies a generative process that provides a mechanism with which to generate samples from the model, it allows for a principled approach to dealing with missing data, a computationally efficient EM algorithm for learning can be derived, and fully Bayesian inference is possible where hyperparameters can be learnt (Bishop, 2006).

The key assumption made in this specification is that the noise is Gaussian, which is a distribution most suited to the analysis of real-valued data. If the data is binary, integer or is non-negative, then this Gaussian assumption is inappropriate. Gaussianity may also be inappropriate for real-valued data that is heavy-tailed. The Poisson distribution is better suited to integer data, the Bernoulli to binary data and the Exponential to non-negative data. A generalisation of PCA that allows the Gaussian assumption to be replaced with a more befitting distribution, would thus be desirable. Such a generalisation is made possible by the fact that many of the distributions of interest in modelling observed data are members of the exponential family of distributions (c.f. section 1.3). The very same motivation has spurred the development of modelling strategies in other statistical settings, most notably in regression with the generalised linear models (GLMs) (Nelder and Wedderburn, 1972). The experience gained with GLMs is brought to bear upon the generalisation of latent variable models.

2.2 Generalising Models to the Exponential Family

2.2.1 Generalised Linear Models

Linear regression is one of the most well-studied of statistical models, relating a set of covariates (features or inputs) $\mathbf{v}_n \in \mathbb{R}^D$ to a set of response variables (labels or outputs) $x_n \in \mathbb{R}$. The relationship between the covariates and the response consists of a systematic component and a random component, described by the linear model:

$$x_n|\mathbf{v}_n \sim \mathcal{N}(x_n|\mu_n(\mathbf{v}_n), \sigma^2), \quad n = 1, \ldots, N \tag{2.5}$$

$$\mathbb{E}[x_n] = \mu_n(\mathbf{v}_n) = \boldsymbol{\beta}\mathbf{v}_n. \tag{2.6}$$

The systematic component $\mu_n = \boldsymbol{\beta}\mathbf{v}_n$ is an approximation to the response variable, often referred to as the 'signal' and the vector $\boldsymbol{\beta} \in \mathbb{R}^D$ is the set of model parameters. The optimal parameters $\boldsymbol{\beta}^*$ are found by minimising the negative log-likelihood, which gives the least squares criterion:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_n (x_n - \boldsymbol{\beta}\mathbf{v}_n)^2 . \tag{2.7}$$

This model remains a key tool for applied statistical work, but has some shortcomings. Consider a problem in which the response variable is integer-valued. An approach to dealing with this setting is to apply a logarithmic transformation to the response variable and thereafter apply the standard linear regression model. This approach fails to take into account the often increasing variance of count-based data with the mean and the discrete nature of the response. The Gaussian assumption, similar to the conclusion of the previous section, is thus undesirable and not generally applicable. In recognition of this shortcoming, models were subsequently developed for binary response regression, polytomous logistic (multinomial) regression and others.

Nelder and Wedderburn (1972) introduced the *generalised linear models* (GLM) by recognising the shared properties that distributions of the exponential family share with each other, and demonstrated the unity of many existing models for regression. For GLMs, the random component is given by an exponential family distribution in the canonical form, rather than the Gaussian form assumed in linear regression. The systematic component $\boldsymbol{\beta}\mathbf{v}_n$, now approximates the natural parameters of the chosen exponential family distribution and equations (2.5) and (2.6) become:

$$x_n|\mu_n \sim \operatorname{Expon}(x_n|g(\mu_n)) = h(x_n)\exp\{g(\mu_n)\,x_n - A(g(\mu_n))\} \tag{2.8}$$

$$\mathbb{E}[x_n] = \mu_n = g^{-1}(\boldsymbol{\beta}\mathbf{v}_n), \tag{2.9}$$

where, for the chosen exponential family, $g(\mu_n)$ are the natural parameters, $g(\cdot)$ is the link function that 'links' the mean parameter space to the natural parameter space, and $A(\cdot)$ is the log-partition function, as described in section 1.3. The negative log-likelihood using equation (2.8) is thus:

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_n A(\boldsymbol{\beta}\mathbf{v}_n) - x_n\,\boldsymbol{\beta}\mathbf{v}_n - \ln h(x_n). \tag{2.10}$$

The optimal parameter values are solved, as before, by minimising the negative log-likelihood.

Example 2.1: ***Linear Regression***

The standard linear regression is obtained by considering the case of the Gaussian distribution that has the log-partition function $A(\eta) = \frac{\eta^2}{2}$, with natural parameter η. Using this form and ignoring constant terms, the equivalence between this maximum likelihood criterion (2.10) and the least squares criterion (2.7) can be seen. □

Example 2.2: ***Logistic Regression***

The equally popular logistic regression for binary responses is recovered from the GLM framework, utilising the Bernoulli distribution whose log-partition function is $A(\eta) = -\ln(1 + \exp(\eta))$ with natural parameters η. The objective is more compactly written using a [-1,1] outcome convention instead of the [0,1] convention using $x^* = 2x - 1$. The resulting negative log-likelihood can be simplified and written as:

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_n \ln\left(1 + \exp(-x_n^* \boldsymbol{\beta}\mathbf{v}_n)\right), \tag{2.11}$$

where $x_n^* = 1$ if $x_n = 1$ and $x_n^* = -1$ if $x_n = 0$. □

The GLM framework provides a mechanism for generalising the least squares regression to loss functions that are more appropriate for members of the exponential family other than the Gaussian. The general strategy that has been described involves:

- Considering a Gaussian likelihood model with a systematic component μ_n.
- Selecting a member of the exponential family most appropriate for the data under study, such as the Bernoulli, Poisson, Gaussian, Gamma, etc.
- Applying a transformation of the systematic components μ_n to natural parameters η_n of the chosen exponential family using a suitable link function, with the canonical link being appropriate most often.

This general strategy can now be used to construct generalised models for many other settings, with this chapter focusing on the generalisation of latent variable models. The approach has been used in other settings, including generalised additive models (Hastie and Tibshirani, 1990), generalised linear mixed models (Breslow and Clayton, 1993), generalised models for survival analysis (Fahrmeir and Tutz, 2001) and generalised linear multi-link models (G^2L^2M) (Gordon, 2002).

2.2.2 PCA for the Exponential Family

We mirror the preceding model development in this section, and examine unsupervised latent variable models in which the covariates $\mathbf{v}_n$ are now *unobserved*

latent variables. Historically, this modelling approach may have been referred to as 'internal analysis' (Bartlett, 1947), but *unsupervised learning* is now the established name for this analysis in machine learning, statistics and many other areas of applied science. Unsupervised learning is immensely important since it is used to build underlying representations of input data and allows us to explore the patterns and structure inherent in data. These representations can then be used to predict future inputs, for decision making, data visualisation or data compression, amongst others. We will demonstrate many of these applications throughout this thesis.

The Gaussian likelihood model that is generalised here is probabilistic PCA, described by equations (2.1) – (2.4). Let $\mathbf{X}$ be an $D \times N$ matrix of observed data, whose nth column is $\mathbf{x}_n$, for $n = 1, \ldots, N$. Let $\mathbf{V}$ be a $K \times N$ matrix of latent variables, where K is the dimensionality of the the latent representation with $K < D$, and columns $\mathbf{v}_n$. $\mathbf{\Theta}$ is a $D \times K$ matrix of parameters whose kth column is $\boldsymbol{\theta}_k$. The matrix $\mathbf{\Pi} = \mathbf{\Theta}\mathbf{V}$ is the $D \times N$ matrix of natural parameters with columns $\boldsymbol{\pi}_n$. The Gaussian assumption used in equation (2.2) is replaced with the more general exponential family distribution with natural parameters $\boldsymbol{\pi}_n$:

$$p(\mathbf{x}_n|\mathbf{v}_n, \mathbf{\Theta}) = \prod_{n=1}^{N} \text{Expon}\left(\mathbf{x}_n|\boldsymbol{\pi}_n\right), \tag{2.12}$$

$$\boldsymbol{\pi}_n = \sum_k v_{nk}\boldsymbol{\theta}_k = \mathbf{\Theta}\mathbf{v}_n. \tag{2.13}$$

The loss function for maximum likelihood parameter learning is thus:

$$\mathcal{L}(\mathbf{V}, \mathbf{\Theta}) = -\ln p(\mathbf{X}|\mathbf{V}, \mathbf{\Theta}) = -\sum_n \ln p(\mathbf{x}_n|\boldsymbol{\pi}_n) \tag{2.14}$$

$$= -\sum_n \left(\mathbf{x}_n^\top \boldsymbol{\pi}_n - A(\boldsymbol{\pi}_n)\right) \tag{2.15}$$

$$= \sum_n B_{A^*}\left(\mathbf{x}_n, g(\boldsymbol{\pi}_n)\right). \tag{2.16}$$

This loss function changes depending on the choice of exponential family most appropriate for the data being studied. The loss function (2.15) follows from the exponential family form, where $A(\cdot)$ is the appropriate log-partition function; constant terms have been omitted. Equation (2.16) follows from the correspondence between the exponential family and the Bregman divergence B_{A^*} (discussed in section 1.3.4), and $g(\cdot)$ is the link function described in section 1.3. This highlights an additional viewpoint from which to understand the learning process, i.e. as the minimisation of a Bregman divergence between the data and its reconstructions. For Gaussian data, the Bregman divergence is the Euclidean distance, and hence corresponds to the usual distance measure used for PCA. This generalisation of PCA is referred to as Exponential Family PCA (EPCA) (Moustaki and Knott, 2000; Collins et al., 2002).

Example 2.3: ***Standard PCA***

The standard PCA is obtained by using equation (2.15) with the Gaussian log-partition function $A(\pi_{nd}) = \frac{\pi_{nd}^2}{2}$. The standard PCA loss (Joliffe, 2002) is:

$$\mathcal{L}_{PCA} = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}_n - g(\boldsymbol{\pi}_n)\|^2, \tag{2.17}$$

where the canonical link function for the Gaussian is the identity, i.e. $g(\pi_{nd}) = \pi_{nd}$. Thus the objective function (2.15) is equivalent to this loss (2.17), ignoring constant terms. This example is an unsupervised analogue of example 2.1 for linear regression. □

Example 2.4: ***Logistic PCA***

Logistic PCA is obtained by employing a Bernoulli likelihood with the logistic link function. The log partition function is $A(\pi_{nd}) = -\ln(1 + \exp(\pi_{nd}))$, giving the loss function:

$$\mathcal{L}_{LPCA}^{bern} = -\sum_{nd} \left(x_{nd}\pi_{nd} + \ln(1 + \exp(\pi_{nd}))\right), \tag{2.18}$$

which is equivalent to the loss function provided by Tipping (1999, eq. 2) and Schein et al. (2003, eq. 4). This example is an unsupervised analogue of example 2.2 for logistic regression. □

Example 2.5: ***Non-negative Matrix Factorisation***

Non-negative matrix factorisation (Lee and Seung, 1999) can also be obtained from the generalisation of PCA discussed here. Exponential family PCA with a Poisson likelihood has a canonical log-partition function $A(\pi_{nd}) = \exp(\pi_{nd})$. The loss functions for NMF (Lee and Seung, 1999, eq. 2) and EPCA are:

$$\mathcal{L}_{NMF} = -\sum_{nd} x_{nd}\ln(\pi_{nd}) + \pi_{nd}, \tag{2.19}$$

$$\mathcal{L}_{EPCA}^{poiss} = -\sum_{nd} x_{nd}\pi_{nd} + \exp(\pi_{nd}). \tag{2.20}$$

The difference between the two loss functions is due to the use of different link functions. The EPCA loss function uses the canonical link, which for the Poisson is the logarithm, whereas NMF makes use of a substitute link function viz. the identity. While both losses (2.19), (2.20) imply Poisson noise, the difference has a bearing on the learning in the model and how the the underlying factors are interpreted in terms of the observed data. The use of the identity link imposes positivity constraints on the model parameters $\boldsymbol{\pi}_n$ and allows for a parts-based interpretation of the NMF factors. The positivity constraint is obviated if the canonical link is used, but linear combinations of factors explain

the generation of the data. The use of substitute link functions is discussed further in section 2.3.3.4. □

Example 2.6: ***Probabilistic Latent Semantic Analysis***

Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) is a model for categorical data that uses a multinomial likelihood. The loss functions for PLSA and EPCA are:

$$\mathcal{L}_{PLSA} = -\sum_{nd} x_{nd} \ln(\pi_{nd}), \tag{2.21}$$

$$\mathcal{L}_{EPCA}^{mult} = -\sum_{nd} x_{nd}\pi_{nd} - \ln\left(\sum_{d} \exp(\pi_{nd})\right). \tag{2.22}$$

PLSA makes use of the identity link, which again requires constraints on the natural parameters to ensure validity. The canonical link is the softmax function and deals with the required constraints automatically. □

The previous two examples highlight the differences between various models based on the use of different link functions – a characteristic that is not often recognised. The examples also highlight an important property of generalised modelling, namely the estimation of parameters in either the natural parameter or mean parameter space. Both NMF and PLSA estimate model parameters that lie in the same space as the observed data, referred to as the mean parameter space; for Bernoulli data the mean parameters are probabilities of being on or off, or for Gaussian data the mean parameters are location values on the same scale as the data. Estimation in the mean parameter space requires constraints to be explicitly handled during learning, e.g leading to the multiplicative updates needed to maintain positivity in NMF. We are not required to manage constraints in generalised latent variable models, because learning is performed in the natural parameter space and constraints are automatically handled through the use of an appropriate link function.

The recognition of the shared properties of the distributions in the exponential family and the potential for the generalisation described above has been recognised by a number of researchers. Two substantial pieces of research in this area are those of Moustaki and Knott (2000), who discuss generalised latent trait models and Collins et al. (2002) who focus on the generalisation of PCA to the exponential family. Our unified presentation hopefully clarifies the link between these various models and adds to the wider discourse in this area. These related works and the contributions of this chapter will be placed within the wider context in section 2.8.

2.2.3 Maximum Likelihood EPCA

Two general approaches for determining maximum likelihood estimates for generalised latent variable models are by Expectation Maximisation (EM) or by direct optimisation. EM is a powerful and highly popular method for determining maximum likelihood solutions in latent variable models (Dempster et al., 1977). In EM, we marginalise the joint likelihood over the latent factors $\mathbf{v}_n$ and maximise over parameters $\boldsymbol{\theta}$. Tipping and Bishop (1997) describe an EM algorithm for probabilistic PCA, which is an effective algorithm for parameter learning since the marginalisation of the latent variables can be done analytically in this linear Gaussian setting. For generalised latent variable models, it is no longer possible to marginalise the latent variables, because the likelihood is no longer conjugate to the Gaussian latent variables. Moustaki and Knott (2000) describe an EM algorithm for generalised latent variable models. They approach the marginalisation of latent variables using numerical integration methods, which has limited accuracy, and were able to demonstrate the method for two latent factors only.

Collins et al. (2002) present a general purpose algorithm for parameter learning in EPCA based on an alternating minimisation procedure. Alternating minimisation algorithms are also known as co-ordinate descent algorithms and are in widespread use, appearing in the early work of Csiszár and Tusnády (1984) and more recently for learning in related work by Zass and Shashua (2006); Lee et al. (2007); Friedman et al. (2007); Mairal et al. (2010). Alternating minimisation procedures, as the naming suggests, are based on alternately optimising the loss function $\mathcal{L}$ with respect to one argument, while keeping all other arguments fixed. Let $\mathbf{V}^{(t)}$ and $\boldsymbol{\Theta}^{(t)}$ represent the parameters at the t^{th} iteration, with $\mathbf{V}^{(0)}$ as a random initialisation. The iterative updates for the EPCA loss function (2.15) are:

$$\boldsymbol{\Theta}^{(t)} = \operatorname*{argmin}_{\boldsymbol{\Theta}} \mathcal{L}(\mathbf{V}^{(t-1)}, \boldsymbol{\Theta}) \tag{2.23}$$

$$\mathbf{V}^{(t)} = \operatorname*{argmin}_{\mathbf{V}} \mathcal{L}(\mathbf{V}, \boldsymbol{\Theta}^{(t)}). \tag{2.24}$$

This approach is amenable to parameter learning in EPCA due to the convex properties of the loss function. The loss function is not convex in the two arguments jointly, but the loss function is convex in either of its arguments with the other fixed. This implies that each iterative update can be done efficiently using the wide array of tools for convex optimisation that are available (Boyde and Vandenberge, 2004). It is unusual to follow a co-ordinate descent algorithm in models with latent variables, since this approach ignores posterior uncertainty in the latent variables and results in overfitting, will be problematic in missing data settings, and can have slow convergence rates. This will also be a poor minimisation scheme if there is high correlation between the latent variables and parameters. Notwithstanding these concerns, we

use this method for our comparisons since the alternating minimisation approach has become the established and popular approach for EPCA learning.

2.3 A Bayesian Exponential Family PCA

2.3.1 Motivating a Bayesian Approach

The maximum likelihood approach to learning discussed thus far has a number of shortcomings that can be addressed by a Bayesian treatment of matrix factorisation models. Firstly, maximum likelihood learning produces point estimates $\{\mathbf{V}^*, \mathbf{\Theta}^*\}$ of the parameters. Ideally though, we wish to learn the posterior distribution $p(\mathbf{V}, \mathbf{\Theta}|\mathbf{X})$ and use this distribution to make predictions of unseen data. Secondly, maximum likelihood estimates are prone to overfitting, resulting in models that fit part of the data perfectly. This is most undesirable since the model will be unable to make predictions of data that it has not been trained with. In these circumstances, resorting to maximum a posteriori (MAP) solutions, where the maximum of the posterior distribution is used instead, seems desirable but does not overcome this problem since the maximum of the posterior is not representative of the entire distribution. MAP solutions are also not invariant to reparameterisation, which detracts from their appeal. Further discussion of these issues is left to the insightful discussion in the books by MacKay (2003); Gelman et al. (2004) and Bishop (2006).

In the case of generalised latent variable models, the maximum a posteriori approach defines a generative process over elements of the observed training matrix, but is ill-posed to predict new rows of the matrix not part of this set, because the latent variables are set to their MAP values. This issue and the theoretical limits of MAP estimation in this setting were brought to light by Welling et al. (2008) for the class of models labelled deterministic latent variable models, of which maximum likelihood EPCA is a member, as well as NMF (example 2.5) and PLSA (example 2.6), and expanded to other cases by (Singh, 2009). The findings of this work are not applicable to Bayesian methods, since a complete generative description over both seen and unseen data elements is specified in all cases.

A Bayesian approach provides a natural framework in which to incorporate prior information into statistical models. The inclusion of the prior provides a built-in regularisation, allowing Bayesian methods to avoid problems with over-fitting. Prior information can include the specification of plausible links between random variables, restrictions on the range of parameter values and probabilistically expressing the underlying process that is believed to generate the observed data. The ability to incorporate prior information makes it possible to extend many models to increasingly complex cases through the use of hierarchical Bayesian modelling

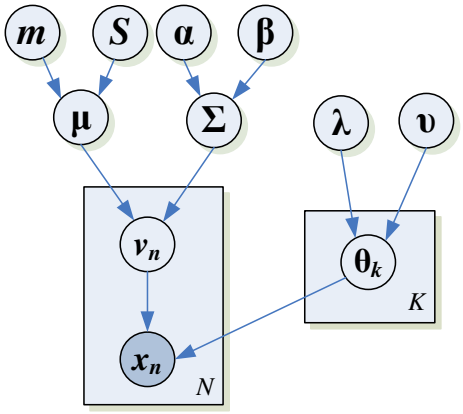


Figure 2.2: Graphical model for Bayesian exponential family PCA.

(Gelman et al., 2004). It is also often the case that better performance can be demonstrated with Bayesian methods than with maximum likelihood methods e.g. Salakhutdinov and Mnih (2008).

Approaches to Bayesian inference provide a mechanism by which to learn the posterior distribution of latent variables and parameters, and thus provides additional motivation for the development of Bayesian models for unsupervised learning. The estimation of these distributions is of interest in a number of application areas, particularly where the latent variables are subject to interpretation and further analysis. Since there is also a great deal of uncertainty in specifying many models, Bayesian methods provide a principled approach for selecting and averaging across plausible models when performing inference and prediction (Bishop, 2006).

2.3.2 Model Construction

We develop a generalised Bayesian latent variable model using the hierarchical model depicted in figure 2.2. The notation used for the specification of EPCA in section 2.2.2 is repeated here for clarity. The shaded node indicates the observed data, which forms as a $D \times N$ matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, with an individual data point $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$. N is the number of data points and D is the number of input features. $\mathbf{\Theta}$ is a $D \times K$ matrix of parameters with columns $\boldsymbol{\theta}_k$. $\mathbf{V}$ is a $K \times N$ matrix of latent variables $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$, with columns $\mathbf{v}_n = [v_{n1}, \ldots, v_{nK}]$ that are K-dimensional vectors of continuous values in $\mathbb{R}$. K is the number of latent factors representing the dimensionality of the sought after underlying representation.

Let $\mathbf{m}$ and $\mathbf{S}$ be hyperparameters representing a K-dimensional vector of mean values and a covariance matrix respectively. Let α and β be the hyperparameters corresponding to the shape and scale parameters of an inverse-Gamma distribution. The model is defined by drawing $\boldsymbol{\mu}$ from a Gaussian distribution and the elements

σ_k^2 of the diagonal matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_K^2)$ from an inverse Gamma distribution:

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}, \mathbf{S}), \tag{2.25}$$

$$\sigma_k^2 \sim \mathcal{G}^{-1}(\alpha, \beta). \tag{2.26}$$

For each data point n, the K-dimensional latent representation $\mathbf{v}_n$ is drawn:

$$\mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_n|\boldsymbol{\mu}, \mathbf{\Sigma}). \tag{2.27}$$

The data is described by an exponential family distribution with model parameters $\boldsymbol{\theta}_k$. The exponential family distribution modelling the data and the corresponding prior over the model parameters is:

$$\mathbf{x}_n|\mathbf{v}_n, \mathbf{\Theta} \sim \text{Expon}\left(\sum_k v_{nk}\boldsymbol{\theta}_k\right), \tag{2.28}$$

$$\boldsymbol{\theta}_k \sim \text{Conj}\left(\boldsymbol{\lambda}, \nu\right). \tag{2.29}$$

The set of parameters to be learnt is $\mathbf{\Omega} = \{\mathbf{V}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$ and the set of hyperparameters is $\mathbf{\Psi} = \{\mathbf{m}, \mathbf{S}, \alpha, \beta, \boldsymbol{\lambda}, \nu\}$. Given the graphical model, the joint probability of all parameters and variables is:

$$p(\mathbf{X}, \mathbf{\Omega}|\mathbf{\Psi}) = p(\mathbf{X}|\mathbf{V}, \mathbf{\Theta})p(\mathbf{\Theta}|\boldsymbol{\lambda}, \nu)p(\mathbf{V}|\boldsymbol{\mu}, \mathbf{\Sigma})p(\boldsymbol{\mu}|\mathbf{m}, \mathbf{S})p(\mathbf{\Sigma}|\alpha, \beta). \tag{2.30}$$

Using the model specification given by equations (2.25) – (2.29), the log-joint probability distribution is:

$$\begin{aligned}
\ln p(\mathbf{X}, \mathbf{\Omega}|\mathbf{\Psi}) = &\sum_{n=1}^{N}\left[\mathbf{x}_n^\top\left(\sum_k v_{nk}\boldsymbol{\theta}_k\right) - A\left(\sum_k v_{nk}\boldsymbol{\theta}_k\right)\right] \\
&+ \sum_{k=1}^{K}\left[\boldsymbol{\lambda}^\top\boldsymbol{\theta}_k - \nu A(\boldsymbol{\theta}_k) - f(\boldsymbol{\lambda}, \nu)\right] \\
&- \sum_{n=1}^{N}\left[\frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{\Sigma}| + \frac{1}{2}(\mathbf{v}_n - \boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\mathbf{v}_n - \boldsymbol{\mu})\right] \\
&- \frac{K}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{S}| - \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{m})^\top\boldsymbol{S}^{-1}(\boldsymbol{\mu} - \boldsymbol{m}) \\
&+ \sum_{k=1}^{K}\left[\alpha\ln\beta - \ln\Gamma(\alpha) + (\alpha - 1)\ln\sigma_k^2 - \beta\sigma_k^2\right],
\end{aligned} \tag{2.31}$$

where the functions $h(\cdot)$, $A(\cdot)$ and $f(\cdot)$ correspond to the functions of the chosen conjugate-exponential family pair of distributions (c.f. Table 1.2). It is also important to note that while conjugate distributions have been used between elements of the model, the model is not wholly conjugate. This model will be referred to by the shorthand BXPCA, referring to Bayesian Exponential Family PCA.

Example 2.7: ***Probabilistic PCA as a Special Case***

The hierarchical construction recovers the familiar probabilistic PCA, discussed in section 2.1, as a special case. This can be shown as in previous examples by considering the the Gaussian-Gaussian conjugate pair as priors for $\mathbf{V}$ and $\mathbf{\Theta}$ using the log-partition function for the Gaussian, and comparing the resulting log-joint likelihood to that of probabilistic PCA given by equation (2.4). □

2.3.3 Properties of the Construction

2.3.3.1 Derivatives of the Likelihood Function

The derivatives of the likelihood function, as well as the full joint probability, will be used for MCMC learning in this Bayesian model. These derivatives are also used in maximum likelihood learning of the model parameters. The derivatives of the likelihood are:

$$\frac{\partial \ln p(\mathbf{X}|\mathbf{V},\mathbf{\Theta})}{\partial \mathbf{V}} = \mathbf{\Theta}^\top \mathbf{X} - \mathbf{\Theta}^\top A'_{\mathbf{V}}(\mathbf{\Theta V}) \tag{2.32}$$

$$\frac{\partial \ln p(\mathbf{X}|\mathbf{V},\mathbf{\Theta})}{\partial \mathbf{\Theta}} = \mathbf{X}\mathbf{V}^\top - A'_{\mathbf{\Theta}}(\mathbf{\Theta V})\mathbf{V}^\top, \tag{2.33}$$

where $A'_{\mathbf{V}}(\mathbf{\Theta V})$ is the derivative of the log-partition function with respect to the matrix $\mathbf{V}$, and similarly for the derivative w.r.t $\mathbf{\Theta}$. These derivatives form a set of coupled equations that can be used in an alternating fashion and are exactly the equations that would be needed in the alternating optimisation for the maximum likelihood solution (section 2.2.3). For the case of Gaussian data, these updates are:

$$\frac{\partial \ln p(\mathbf{X}|\mathbf{V},\mathbf{\Theta})}{\partial \mathbf{V}^{(t)}} = \mathbf{\Theta}^{(t-1)\top}(\mathbf{X} - \mathbf{\Theta}^{(t-1)}\mathbf{V}^{(t)}) \tag{2.34}$$

$$\frac{\partial \ln p(\mathbf{X}|\mathbf{V},\mathbf{\Theta})}{\partial \mathbf{\Theta}^{(t)}} = (\mathbf{X} - \mathbf{\Theta}^{(t)}\mathbf{V}^{(t)})\mathbf{V}^{(t)\top}, \tag{2.35}$$

where the solutions obtained at iteration t are denoted by $\mathbf{V}^{(t)}$ and $\mathbf{\Theta}^{(t)}$. Equating (2.34) and (2.35) to zero and substituting the update for $\mathbf{V}$ into $\mathbf{\Theta}$ gives:

$$\mathbf{\Theta}^{(t)} = \frac{1}{C}\mathbf{X}\mathbf{X}^\top\mathbf{\Theta}^{(t-1)}, \tag{2.36}$$

where C is a scalar. $\mathbf{\Theta}$ is the basis of the underlying subspace and corresponds to the set of principal components. The update (2.36) is equivalent to the power method for determining the eigenvector of $\mathbf{X}\mathbf{X}^\top$ with the largest eigenvalue (Golub and Van Loan, 1996, pp. 330). This is the best one-component solution for $\mathbf{\Theta}$ and provides a link to one of the classical methods for solving the standard PCA problem.

2.3.3.2 Mixture Interpretation

A common strategy in unsupervised modelling involves the marginalisation over latent variables. Employing this strategy using equation (2.28) results in:

$$\begin{aligned} p(\mathbf{x}_n|\boldsymbol{\Theta}) &= \int p(\mathbf{v}_n|\boldsymbol{\mu},\boldsymbol{\Sigma})p(\mathbf{x}_n|\mathbf{v}_n,\boldsymbol{\Theta})d\mathbf{v}_n \\ &= \int \mathcal{N}(\mathbf{v}_n|\boldsymbol{\mu},\boldsymbol{\Sigma})\mathrm{Expon}\left(\mathbf{x}_n|\boldsymbol{\Theta}\mathbf{v}_n\right) d\mathbf{v}_n. \end{aligned} \tag{2.37}$$

The observed data $\mathbf{x}_n$ is effectively being modelled as a Gaussian mixture of exponential family distributions. This gives insight and guidance for learning parameters of the model. If the exponential family distribution under study is Gaussian, the mixture is Gaussian. Efficient inference can be performed by recognising this property, and is the strategy employed for learning in probabilistic and Bayesian PCA (Tipping and Bishop, 1997; Bishop, 1999). For other distributions in the exponential family, a different distributional form is obtained when marginalising over the latent variables, which has a bearing on learning in the model. To effectively explore the posterior distribution in this setting, we make use of Hybrid Monte Carlo sampling, which uses gradient information to aid the exploration of the posterior.

2.3.3.3 Aspects of Model Identifiability

Identifiability of model parameters is a concern in many applications of latent variable models, particularly in cases where the researcher aims to provide an interpretation for the factors that are learnt. In general, inferred factors are statistical quantities and do not have any physical basis, but an interpretation is often made in practice on the basis of posterior summaries of latent variables. For unidentified parameters, this summarisation is not possible. In the exponential family PCA model, the product $\boldsymbol{\Pi} = \boldsymbol{\Theta}\mathbf{V}$ is identified but $\mathbf{V}$ and $\boldsymbol{\Theta}$ are not, since for any orthogonal matrix $\mathbf{R}$, $\boldsymbol{\Theta}\mathbf{V} = (\boldsymbol{\Theta}\mathbf{R}^\top)(\mathbf{R}\mathbf{V})$. For problems of prediction, missing data imputation and data reconstruction, the lack of identifiability (also called factor indeterminacy) is not an obstacle, since the (identified) product $\boldsymbol{\Theta}\mathbf{V}$ can be computed for all samples and thereafter averaged to obtain the probabilities for individual data points.

There are two general strategies that can be used to ensure identifiability if this is an aspect of the model design. The first broad set of strategies is to impose *constraints* on the loadings matrix $\boldsymbol{\Theta}$, with such constraints often suggested by the application. Since latent variables are often introduced for convenience, one approach is to set the upper triangular elements of $\boldsymbol{\Theta}$ to zero, following the specification of Geweke and Zhou (1996) and demonstrated by other authors such as Lopes and West (2004). The disadvantages of such constraints are that they change the model and make learning more difficult. The second class of approaches for handling

Table 2.1: Substitute link functions for four distributions. The canonical link functions are indicated by red squares in the tables. $\Phi^{-1}(\cdot)$ is the inverse Gaussian CDF.

Link Name	Function	Bernoulli	Poisson	Gaussian	Gamma
Identity	ω		★	■	★
Reciprocal	$1/\omega$			★	■
Square Root	$\sqrt{\omega}$		★		
Log	$\log(\omega)$	★	■	★	★
Logit	$\log\left(\frac{\omega}{1-\omega}\right)$	■			
Probit	$\Phi^{-1}(\omega)$	★			
Complementary log-log	$\log(-\log(\omega))$	★			

identifiability, is to introduce additional *post-processing* steps after parameter learning in the non-identified model. The post-processing strategy is explored further in section 2.5.

2.3.3.4 Substitute Link Functions

The canonical link function is often the most appropriate link function for a wide range of applications. It is possible to use a non-canonical or *substitute link function*, thus employing a reparameterised, non-canonical exponential family in learning. This is especially important in certain generalised learning settings, one particular case being the learning of non-negative data using a Gamma likelihood. The canonical link function for the Gamma distribution is the reciprocal i.e. $\omega = -\frac{1}{\eta}$, where ω is the Gamma distribution's scale parameter, and η being the natural parameter. The requirement that the scale $\omega > 0$, thus imposes a negativity constraint on the natural parameters.

The Bayesian exponential family PCA (BXPCA) model as specified will not satisfy this negativity constraint in general, requiring some adjustment of the model to meet this requirement. The approach taken by Moustaki and Knott (2000) is to specify the model with latent variables that are constrained Gaussians. This is not generally desirable, particularly in the case where mixed data is considered, such as mixed data of binary and non-negative observations. In such a setup, the latent variables will be constrained for *all* parameters, which is unnecessary. An alternative solution is to make use of a substitute link function. Substitute link functions are often known for many distributions of interest, such as those listed in table 2.1. The effect of using substitute link functions on the maximum likelihood objective function was also discussed for EPCA in the examples of section 2.2.2. For the Gamma distribution, the logarithmic-link is constraint-free and is thus appropriate for use in modelling non-negative data with the Gamma distribution. The use of non-canonical link functions results in curved rather than regular exponential families (Bickel and Doksum, 2001, pp. 416) and their use has been widely studied

for generalised linear models. For GLMs, canonical links are preferred in general (Bickel and Doksum, 2001).

2.3.4 Posterior Computation

Learning in the Bayesian exponential family framework involves sampling all unknown variables, denoted by the set $\mathbf{\Omega} = \{\mathbf{V}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$, given the observed data. The top level parameters in figure 2.2, $\mathbf{\Psi} = \{\mathbf{m}, \mathbf{S}, \alpha, \beta, \boldsymbol{\lambda}, \nu\}$ are treated as fixed hyperparameters, but these can be learnt from the data. Since all parameters of interest are continuous, it is possible to compute derivatives of the log-joint probability $p(\mathbf{X}, \mathbf{\Omega}|\mathbf{\Psi})$. This property, coupled with the the earlier observation regarding the need for an effective sampling scheme due to the potential sensitivities in learning, makes Hybrid (sometimes called Hamiltonian) Monte Carlo an appealing sampling approach. Hybrid Monte Carlo (HMC) was described in section 1.5.2 and makes use of gradient information to aid sampling from the posterior distribution. The additional gradient information helps to overcome the random walk behaviour experienced by other sampling schemes such as Metropolis-Hastings and can lead to dramatically improved mixing of the Markov chain. The potential energy function required for the HMC sampling is $\mathcal{E}(\mathbf{\Omega}|\mathbf{\Psi}) = -\ln p(\mathbf{X}, \mathbf{\Omega}|\mathbf{\Psi})$.

The use of the exponential family form ensures that inference is performed in the space of natural parameters and not the original data or mean parameter space. This natural parameter representation allows sampling of the matrices $\mathbf{V}$ and $\mathbf{\Theta}$ to be done in an unconstrained space, which makes inference in general easier and is particularly useful for HMC sampling. HMC is also useful since it allows for sampling in non-conjugate models, of which the model developed here is an example.

The general approach for using HMC with constrained variables was described in section 1.5.2.1. The only constrained variable in the model is $\mathbf{\Sigma}$, where each diagonal element $\sigma_k^2 > 0$. Each σ_k^2 can be transformed to a corresponding unconstrained variable ξ_k using the transformation: $\sigma_k^2 = \exp(\xi_k)$. This transformation requires that the chain rule for differentiation is applied and that the determinant of the Jacobian of the transformed variables be included.

Example 2.8: ***Binary Matrix Factorisation Model***

It is illustrative to consider a model for binary data using the Beta-Bernoulli conjugate-exponential pair. The Jacobian for the variance transformation is:

$$|\mathbf{J}| = \left| \frac{\partial}{\partial \xi_k} \exp(\sigma_k^2) \right| = |\exp(\xi_k)| = \sigma_k^2. \tag{2.38}$$

The final Potential energy function, which includes the Jacobian term can then be written as:

$$\begin{aligned}\ln p(\mathbf{X}, \boldsymbol{\Omega}|\boldsymbol{\Psi}) = &\sum_{n=1}^{N} \mathbf{x}_n^\top \left(\sum_k v_{nk} \boldsymbol{\theta}_k \right) - \sum_{d=1}^{D} \ln \left(1 + \exp \left\{ \sum_k v_{nk} \theta_{kd} \right\} \right) \\ &+ \sum_{k=1}^{K} \left[\sum_{d=1}^{D} \left(-\lambda_1 \ln(1 + e^{-\theta_{kd}}) - \lambda_2 \ln(1 + e^{\theta_{kd}}) \right) \right. \\ &\left. - \sum_{d=1}^{D} (\theta_{kd} - 2 \ln(1 + e^{\theta_{kd}}) + D \ln \frac{\Gamma(\lambda_1 + \lambda_2)}{\Gamma(\lambda_1)\Gamma(\lambda_2)} \right] \\ &- \sum_{n=1}^{N} \left(\frac{K}{2} \ln(2\pi) + \frac{1}{2} \ln |\boldsymbol{\Sigma}| + \frac{1}{2} (\mathbf{v}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{v}_n - \boldsymbol{\mu}) \right) \\ &- \frac{K}{2} \ln(2\pi) - \frac{1}{2} \ln |\boldsymbol{S}| - \frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{m})^\top \boldsymbol{S}^{-1} (\boldsymbol{\mu} - \boldsymbol{m}) \\ &+ \sum_{k=1}^{K} \left[\alpha \ln \beta - \ln \Gamma(\alpha) + (\alpha - 1) \ln \sigma_k^2 - \beta \sigma_k^2 + \underbrace{\ln \sigma_k^2}_{|J|} \right]. \end{aligned} \tag{2.39}$$

The required derivatives for HMC can now be computed using this expression and differentiating with respect to each of the variables in the set $\boldsymbol{\Omega}$. □

The HMC procedure is implemented to handle missing inputs in a principled manner. The data is divided into the set of observed and missing data, $\mathbf{X} = \{\mathbf{X}^{obs}, \mathbf{X}^{missing}\}$, and the set $\mathbf{X}^{obs}$ is used for inference. In practice, the pattern of missing data is represented by a masking matrix, which is an indicator matrix representing elements that are observed versus missing. Probabilities are then computed using the elements of the masking matrix set to one.

The exponential family representation allows for the modelling of heterogeneous data in a single framework. The evaluations shown in this chapter assume that all data is of the same type, but learning of mixed data types, where some features are integers and others binary for example, can easily be accommodated by representing some of the elements of $\boldsymbol{\Theta}$ as parameters of the Poisson distribution and the remaining elements as parameters of a Bernoulli distribution respectively.

2.4 Evaluating Model Performance

2.4.1 Testing Methodology

We evaluate the exponential family model developed using both synthetic and real world data. We define training and testing data for each of the available data sets. The test data is chosen by randomly selecting 10% of the elements of $\mathbf{X}$. These test

elements are represented as missing data in the training data set and we learn in the presence of missing data. Twenty such data sets are created, each with a different set of missing data and we report the mean and standard deviation error bars for each of the evaluation metrics used. We use this methodology in all the evaluations presented in this thesis.

For training and testing data $\mathbf{x}_n^{train}$ and $\mathbf{x}_n^{test}$ respectively for $n = 1, \ldots, N$, the algorithms under study are evaluated using the root mean squared error (RMSE) and the predictive probability (NLP). The RMSE is evaluated as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_n (\mathbf{x}_n^{test} - \mathbf{x}_n^{pred})^2}. \tag{2.40}$$

The RMSE makes most sense for Gaussian data, but is commonly used in other settings. The negative log-predictive probability (NLP), sometimes referred to as the test likelihood or expected deviance is:

$$\begin{aligned} \text{NLP} &= -\ln p\left(\mathbf{x}^{test}|\mathbf{x}^{train}\right) \\ p\left(\mathbf{x}^{test}|\mathbf{x}^{train}\right) &= \int p\left(\mathbf{x}^{test}|\mathbf{\Omega}\right) p\left(\mathbf{\Omega}|\mathbf{x}^{train}\right) d\mathbf{\Omega}, \end{aligned} \tag{2.41}$$

where the last equation is computed by Monte Carlo evaluation of the integral using samples $\mathbf{\Omega}^{(s)}$ drawn from the posterior distribution, which are sampled during the learning process. A wider discussion of metrics for model checking and comparison is given in the book by Gelman et al. (2004, pp. 180).

2.4.2 Binary Synthetic Data Analysis

Consider a model for binary data based on the Beta-Bernoulli model considered in example 2.8. Synthetic data was generated by creating three 16-bit prototype vectors, with each bit being generated with a probability of 0.5. Each of the three prototypes is replicated 200 times, resulting in a 600-point data set. Bits in the replicates were then flipped with a probability of 0.1, as in Tipping (1999), thus adding noise about each of the prototypes. BXPCA inference was conducted using this data for 6000 iterations of hybrid Monte Carlo, using the first half as burn-in. Figure 2.3 demonstrates the learning process of BXPCA. In the initial phase of the sampling, the model is unable to learn any useful structure from the data (samples 5, 15). The energy function rapidly decreases and some useful structure has been learnt by sample 50. By sample 6000 the model has learnt the original data well, as can be seen by comparing the reconstructions at sample 6000 and the original data.

The rapid evolution of the samples is an indicator of good mixing of the Markov chain. In addition, convergence of the chain is examined using two quantitative

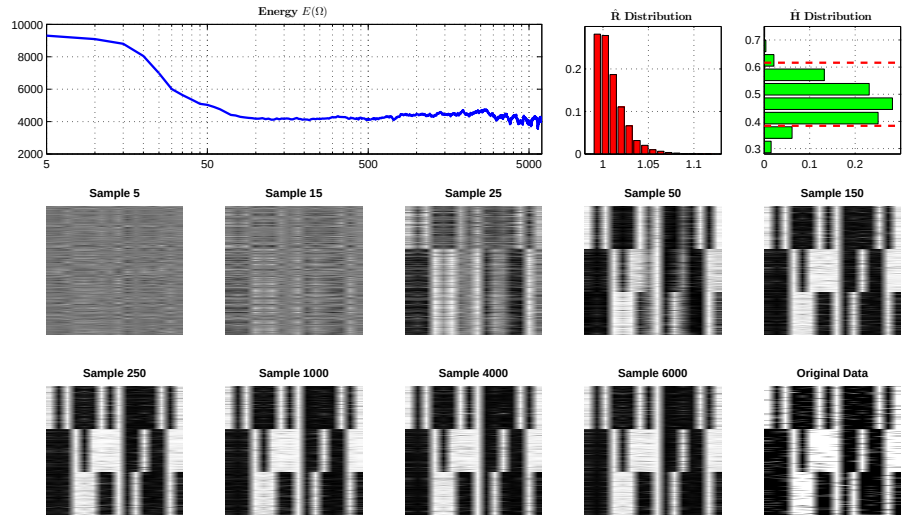


Figure 2.3: Reconstruction of data from samples at various stages of the sampling. The top plot shows the change in the energy function. $\hat{R}$ and $\hat{H}$ are measure of the chain convergence (discussed in text). The lower plots show the greyscale reconstructions and the original data.

convergence diagnostics: the potential scale reduction factor $\hat{R}$, and Brooks's hairiness index $\hat{H}$, that were discussed in section 1.5.4. These tests are evaluated on elements of the reconstruction product $\mathbf{\Theta V}$. The potential scale reduction factor was computed by simulating five separate chains, each with random initialisations. The general rule of thumb is to seek $\hat{R} < 1.1$ (Gelman et al., 2004), which indicates that the chain has been run long enough. The histogram of $\hat{R}$ values in figure 2.3 shows all measurements being less than this cut-off and gives no indication that convergence is an issue. The hairiness index is computed for all samples of a single chain and highlights convergence issues when sample values lie outside the 95% confidence bounds of the test. A histogram of the hairiness indices for elements of the reconstruction product is also shown in figure 2.3, along with the 95% confidence bound. By this test, over 90% of the measurements lie within bounds. The $\hat{H}$ and $\hat{R}$ indicators, in combination with the rapid mixing of the chain give no reason to suspect issues of sampler convergence, providing a high level of trust in the use of the samples for further analysis. Such an analysis can be used for all data sets being evaluated and can be useful in tuning the samplers that are used.

In figure 2.4a and 2.4b, the RMSE of the two algorithms on the training and testing data respectively, are compared for various choices of the latent dimensionality K. EPCA shows underfitting for $K = 1$ and demonstrates severe overfitting for large K. This overfitting is clearly seen in the training data RMSE for EPCA, which quickly goes to zero for larger K, whereas BXPCA manages to avoid this problem. Figure 2.4c shows the NLP of the two methods. A random model is expected to have an $NLP = 10\% \times 600 \times 16 = 960$ bits or normalised to 1.6 bits per observation, but the NLP values for EPCA are significantly larger than this. This is because EPCA

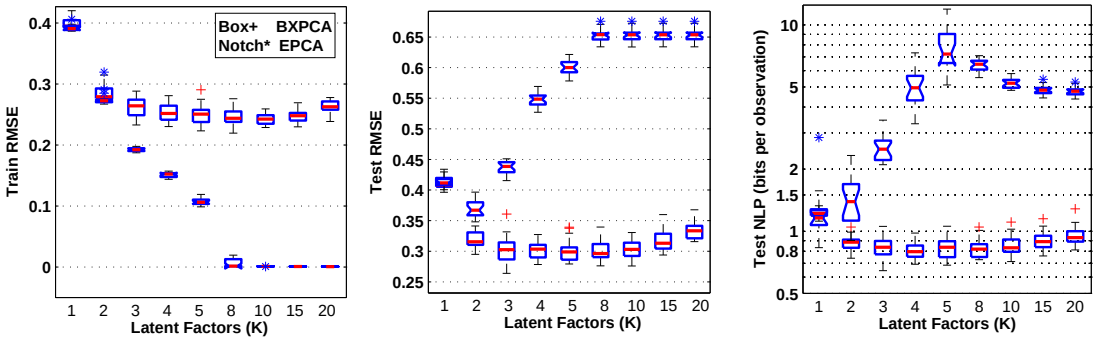


Figure 2.4: Comparison of performance for various latent factors. BXPCA indicated by boxes with '+' for outliers, and EPCA given by notched boxes with '*' for outliers. (a) RMSE on training data (b) RMSE on test data (c) NLP (shown on a log-scale to aid viewing).

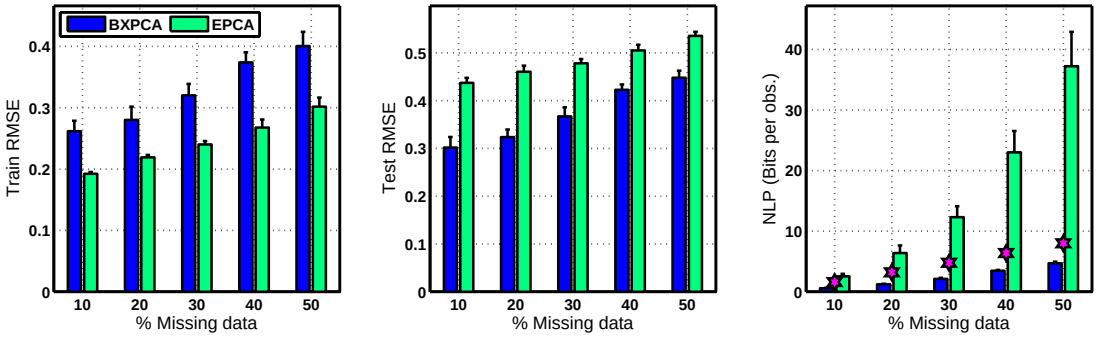


Figure 2.5: Bar plots comparing performance for missing data levels from 10% - 50%.

tends to fit the training data exactly and becomes overconfident in its predictions.

The performance results were shown for an induced 'missingness' level of 10%. The performance of BXPCA and EPCA was compared for $K = 3$ latent factors for various levels of induced missingness, ranging from 10% to 50%, to expose the behaviour of both methods under the various missing data conditions. The results are shown in figure 2.5. The stars in the last plot indicate the NLP for a random predictor. For BXPCA, the increasing level of uncertainty is reflected in all three graphs, showing an increasing trend in the three error measures used. BXPCA is able to provide predictive ability even in settings with high missing data levels, with the NLP for all test scenarios lower than the NLP under a random predictor. EPCA provides a better fit to the training data, but is then unable to provide useful predictions of the unseen data as seen in both the test RMSE and NLP. This analysis conducted shows that Bayesian learning in this model framework provides a mechanism by which to obtain robust inferences from data and allows effective predictions to be made under many varying conditions.

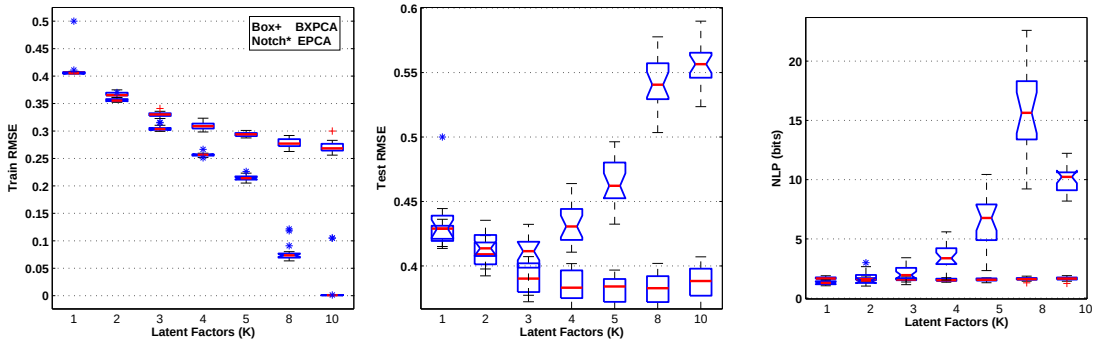


Figure 2.6: Comparison of RMSE and NLP for various latent dimensions for the SPECT images data set. BXPCA indicated by boxes with '+' for outliers, and EPCA given by notched boxes with '*' for outliers. (a) RMSE on training data (b) RMSE on test data (c) NLP.

2.4.3 SPECT Image Analysis

Single Proton Emission Computed Tomography (SPECT) images are used in the diagnosis of abnormal cardiac function. The data used here consists of SPECT images of 267 patients, which has been processed to extract 22 binary attributes that describe the images (UCI Data). Figure 2.6 compares the performance of BXPCA and EPCA. The two algorithms perform equally well with small latent dimensionality K. As the latent dimensionality increases, EPCA begins to over-fit the data, as seen in the plot of training error with a corresponding degradation in the imputation of the unseen test data. The results shows lower error on the testing data for the Bayesian approach. The results also suggest that a latent dimensionality of 4 or 5 is suitable to accurately represent this data.

2.5 Selecting a Final Embedding

The lack of parameter identifiability, discussed in section 2.3.3, poses a problem for certain analyses of the posterior samples obtained. In maximum likelihood methods, the alternating minimisation returns a single $\mathbf{V}$ that is the low dimensional representation. In the Bayesian approach, a single representative for $\mathbf{V}$ is not obtained, but rather many samples, which represent the variation in the embedding. The lack of identifiability subjects $\mathbf{V}$ to permutations of the columns and to rotations of the matrix, making an average of the samples of $\mathbf{V}$ meaningless. This is a problem encountered in many areas of statistical analysis: in mixture modelling this problem is referred to as the 'label switching' problem (Redner and Walker, 1984) or the 'alignment' problem in factor analysis (Clarkson, 1979).

A general strategy by which to induce identifiability in factor models is to constrain model parameters such that symmetries are removed. This is achieved

by imposing constraints on the model parameters, as noted in section 2.3.3 or by post-processing. Post-processing does not affect identifiability since this is a model property, but allows the set of samples obtained from the model without identifiability constraints to be adapted and used to make meaningful inferences. For post-processing, the simplest strategy involves aligning factors based on means, variances or other statistics of interest. Other more advanced relabelling or alignment algorithms also exist, such as those discussed by Stephens (2000). The approach taken here considers the use of further sampling steps, producing a set of post hoc samples, which will allow meaningful averages to be taken.

As an initial approach, a representative embedding can be obtained by choosing the best global configuration from the set of available samples $\{\mathbf{V}^*, \boldsymbol{\Theta}^*\} = \text{argmax}_{\boldsymbol{\Omega}^{(s)}} p(\mathbf{X}, \boldsymbol{\Omega}^{(s)}|\boldsymbol{\Psi})$, and using this $\mathbf{V}^*$ or $\boldsymbol{\Theta}^*$ in any subsequent analysis. This approach does not consider the uncertainty in the embedding obtained and is thus not a method of choice. A second approach aims to give further information about the variability of the embedding. Here, the model parameters $\{\boldsymbol{\Theta}^*, \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\}$ are fixed in order to obtain the embedding for $\mathbf{V}$. These fixed parameters can be set using the sample chosen in the first approach. The embedding $\mathbf{V}$ is then sampled from the conditional distribution:

$$\mathbf{V} \sim p(\mathbf{V}|\mathbf{X}, \boldsymbol{\Theta}^*, \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \propto p(\mathbf{X}|\mathbf{V}, \boldsymbol{\Theta}^*)p(\mathbf{V}|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \tag{2.42}$$

where equation (2.42) is obtained using Bayes' theorem and the joint probability distribution (2.30). Samples are obtained by any preferred MCMC sampling scheme. Problems of rotation and permutation have been removed by constraining the variables $\{\boldsymbol{\Theta}^*, \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\}$ and the ergodic average of the post hoc samples can now be correctly computed. The same procedure can be applied to obtain a representation of the factor loadings $\boldsymbol{\Theta}$, where samples are drawn from the conditional distribution:

$$\boldsymbol{\Theta} \sim p(\boldsymbol{\Theta}|\mathbf{X}, \mathbf{V}^*) \propto p(\mathbf{X}|\mathbf{V}^*, \boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\lambda}, \nu). \tag{2.43}$$

Resampling both $\mathbf{V}$ and $\boldsymbol{\Theta}$ in this way gives an understanding of the variability of the final embedding, in terms of both $\boldsymbol{\Theta}$ and $\mathbf{V}$.

This procedure is demonstrated using the synthetic data described in the previous section for $K = 2$ latent dimensions. A visualisation of the latent factors $\mathbf{V}$ depicts observations that are similar, whereas the visualisation of the factor loadings $\boldsymbol{\Theta}$ depicts similarity between the feature dimensions. Figure 2.7 is a visualisation of the embedding in the two-dimensional space for 10 data points and 20 independent samples drawn for the latent variables $\mathbf{V}$ and for the factor loadings $\boldsymbol{\Theta}$, using equations (2.42) and (2.43). The colours and shapes indicate different observations, where all samples corresponding to the same observation are plotted with the same

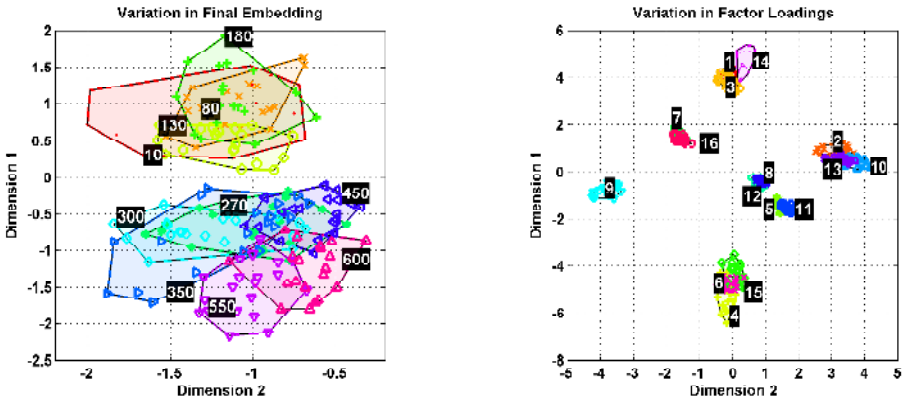


Figure 2.7: Variation in embedding obtained using 20 post hoc samples for (a) embedding of 10 observations of $\mathbf{V}$ and (b) the variation in the factor loadings $\mathbf{\Theta}$ for all 16 dimensions.

shape and colour. The convex hull of each set of samples is also shown by the connecting lines, with the enclosed region shaded for ease of visualisation.

From 2.7(a), two clustered regions can be seen, which represent observed data points that are similar to each other. While three clusters are present in the observed data, two of the three clusters are very similar and this two-dimensional visualisation is unable to separate these two classes. Similarly, figure 2.7(b) shows the similarity of the input dimensions. Dimensions 2, 10 and 13 overlap in figure 2.7(b), and these are highly similar input dimensions, which can be visually supported by examining the input data (shown in the last panel of figure 2.3).

When fixing $\{\mathbf{\Theta}^*, \boldsymbol{\mu}^*, \mathbf{\Sigma}^*\}$ for the resampling of $\mathbf{V}$, it is better to choose a sample randomly from the set of samples at convergence, since choosing the best sample will introduce a bias that can undermine performance. This can also be done for five samples to get an indication of the variation in the embedding in terms of both parameters. Since there is a dependence between $\mathbf{V}$ and $\mathbf{\Theta}$, a high correlation between these two parameters will also result in poor resampling. One way of resolving these concerns would be to follow an EM approach for determining the final embedding, and is appropriate for this visualisation task.

2.6 Study: Elicitation of Scotch Whiskey Preferences

The following case study highlights elements of the practical application of exponential family factor models. Exploratory data analysis is usually the first step in much of applied statistical work, and the exponential models discussed extend the ability to visualise and explore the many diverse data types now available - analysis often restricted to real-valued data. One application of particular interest is

Table 2.2: Summary of the Scotch whiskey data (Edwards and Allenby, 2003).

#	Symbol	Brand	# Users	Price	Bottled	Type
1	CHR	Chivas Regal	806	21.99	Abroad	Blend
2	DWL	Dewar's White Label	517	17.99	Abroad	Blend
3	JWB	Johnnie Walker Black Label	502	22.99	Abroad	Blend
4	JaB	J&B	458	18.99	Abroad	Blend
5	JWR	Johnnie Walker Red Label	424	18.99	Abroad	Blend
6	OTH	Other brands	414			
7	GLT	Glenlivet	354	22.99	Abroad	Single malt
8	CTY	Cutty Sark	339	15.99	Abroad	Blend
9	GFH	Glenfiddich	334	39.99	Abroad	Single malt
10	PCH	Pinch (Haig)	117	24.99	Abroad	Blend
11	MCG	Clan MacGregor	103	10	US	Blend
12	BAL	Ballantine	99	14.9	Abroad	Blend
13	MCL	Macallan	95	32.99	Abroad	Single malt
14	PAS	Passport	82	10.9	US	Blend
15	BaW	Black & White	81	12.1	Abroad	Blend
16	SCY	Scoresby Rare	79	10.6	US	Blend
17	GRT	Grant's	74	12.5	Abroad	Blend
18	USH	Ushers	67	13.56	Abroad	Blend
19	WHT	White Horse	62	16.99	Abroad	Blend
20	KND	Knockando	47	33.99	Abroad	Single malt
21	SGT	Singleton	31	28.99	Abroad	Single malt

in emerging areas of so-called 'algorithmic marketing' or 'computational advertising'.

The Simmons study of media and markets (1997) (Edwards and Allenby, 2003) was conducted to query households regarding brand awareness and product usage. One segment of the study focused on the consumption of Scotch whiskey. The data collected consists of $N = 2218$ respondents and binary indicators of whether or not respondents had bought any of $D = 21$ brands of Scotch over the last year. Table 2.2, lists the brands considered, the number of users, pricing, whether the whiskey is blended or single malt and the bottling location.

This data set was analysed using the Bayesian exponential family PCA (BX-PCA) model with $K = 2$ latent factors as an initial analysis of the data. The latent variables $\mathbf{V}$ represent *user* preferences amongst the the K underlying factors and $\mathbf{\Theta}$ represents the extent to which each of the Scotch *brands* appeal to the various user preferences. The latent factors are expected to reflect factors which affect users' purchasing decisions, such as affordability and reputation. Figure 2.8 provides a view of the data used, where the abbreviations used are listed in table 2.2. The figure also shows hairiness plots for 4 model parameters as a check on mixing properties of the sampler, with curves lying within the 95% confidence intervals.

The aim of the study here is to highlight the potential insights that can be gained for marketing purposes using this modelling approach. One popular area is that of collaborative filtering, which is an information filtering approach which can

be used to make product recommendations to users based on the behaviour of other individuals with similar tastes, such as the popular Netflix challenge (Netflix, 2009). The problem set-up for collaborative filtering is a prediction task of the kind demonstrated in the previous section, and will thus not be explored here further, though is very relevant. Here, the focus will be on more traditional marketing approaches, looking at opportunities for campaign design and the insights for campaigns that can be obtained. A campaign is usually a particular targeting strategy aimed at a choice of predefined users, or a wider choice of advertising aimed at particular sets of users.

2.6.1 Product-space Analysis

Figure 2.8 provides a spatial characterisation of the Scotch brands by showing a plot of the embedding variation for the factor loadings $\mathbf{\Theta}$. The post-processing method described in section 2.5 for selecting a final embedding was used, with the latent representation obtained being similar to the result produced by Edwards and Allenby (2003) using PCA. This representation shows interesting groupings of the various brands by both market share (as indicated by the number of users listed in table 2.2) and the blend of the whiskey. The top 9 brands by usage are clearly distinguishable from the remaining brands (forming a grouping on the left side of the plot). The single malt whiskeys can also be easily identified (bottom right corner of the figure). Dimension 2 is a factor that can be interpreted as the popularity of the Scotch, with whiskeys being raked from most popular on the left (CHR) to least popular on the right (SGT).

2.6.2 User-space Analysis

A latent representation is obtained for every user in the data set, which allows the common behaviour of users to be studied. Figure 2.9 shows a sample from the model for the latent user-space $\mathbf{V}$. A number of interesting features can be observed. There are a number of clusters of Scotch drinkers, which have been highlighted and data contributions for those users shown in the figure insets. The first grouping are those that are consumers of CHR and DWL only. The second group are connoisseurs of single malt Scotch (GLT, GFH) and the third group are those that focus on brands of Scotch 'other' than the widely available options. The marketing analyst would then construct campaigns for targeted advertising on groups of users, who have been selected not simply because they have bought the same brands of Scotch, but because they share the same underlying preferences. This thinking focuses on the 'up-sell' of products (selling more of the same). The collaborative filtering approach combines the view of the users with the spatial characterisation of Scotches to suggest related brands of interest - thereby focusing on the 'cross-sell' aspect of marketing.

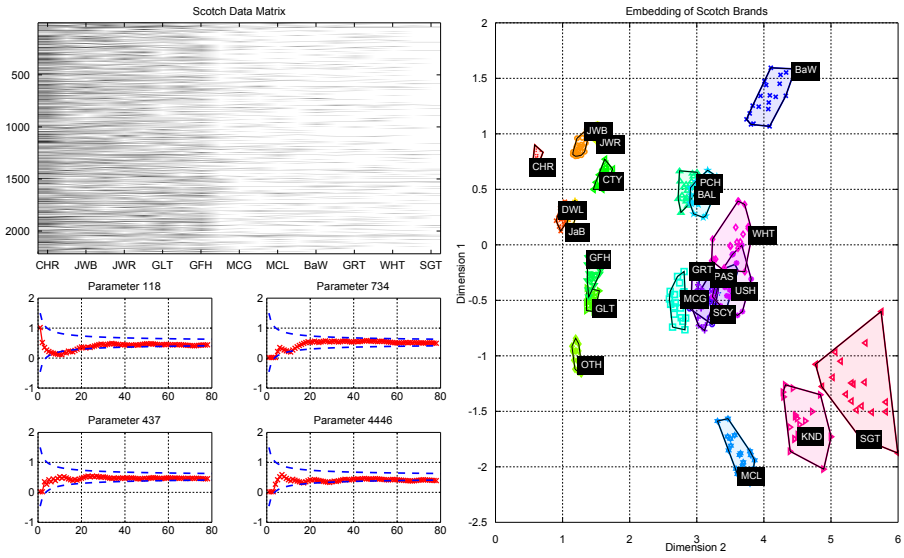


Figure 2.8: Plot showing the Scotch data matrix (top left panel), Hairiness plots (bottom left panel), and the two-dimensional embedding of Scotch brands (right panel).

Current trends will continue to see increased relevance of the modelling techniques discussed here, becoming embedded in current competitive strategies for marketing in both online and shop-font settings. A more sophisticated analysis would involve the inclusion of other sources of data, with the 'Matchbox' model a good example (Stern et al., 2009).

2.7 Methods for Approximate Inference

While we have focussed on MCMC methods throughout this chapter, other approximate inference methods can be used and we contextualise their use here. The first approximate inference method we consider is *variational inference*. In the variational approach, we define the variational free energy (Beal, 2003) of the BXPCA model (here leaving out hyperparameters for simplicity) as:

$$\mathcal{F}(Q(\mathbf{V}, \mathbf{\Theta})) = \mathbb{E}_{Q(\mathbf{V},\mathbf{\Theta})} \left[\ln p(\mathbf{X}, \mathbf{V}, \mathbf{\Theta}) - \ln Q(\mathbf{V}, \mathbf{\Theta}) \right]. \tag{2.44}$$

This variational free energy can be shown to be a lower bound on the log-likelihood $p(\mathbf{X})$ for all distributions $Q(\mathbf{V}, \mathbf{\Theta})$. The variational approximate inference procedure is obtained by maximising $F(Q(\mathbf{V}, \mathbf{\Theta}))$ subject to the variational approximation $Q(\mathbf{V}, \mathbf{\Theta}) = Q(\mathbf{V})Q(\mathbf{\Theta})$. To implement the inference procedure, we must be able to compute expectations with respect to the Q-distributions. The maximisation is achieved by optimising the free energy with respect to $Q(\mathbf{V})$, keeping the $Q(\mathbf{\Theta})$ fixed, and alternating in this way by optimising one keeping the other fixed until

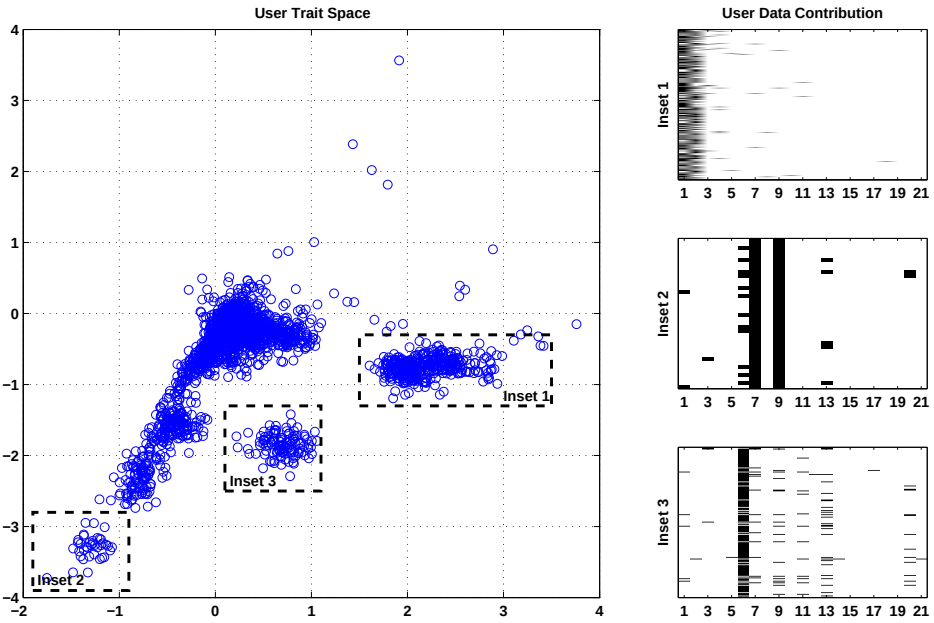


Figure 2.9: Analysis of the latent user trait space. Inset 1, highlights users highly loyal to brands 1 and 2, Inset 2 are single malt connoisseurs and Inset 3 are 'other' Scotch drinkers.

convergence. Variational methods take into account the whole posterior distribution, and unlike MAP estimates, can avoid overfitting in this way.

Variational inference for the standard factor analysis model was shown by Ghahramani and Beal (2000). In the Bayesian exponential family PCA model, the required expectations are more difficult to compute, since we must take expectations of the log-partition function. To overcome this difficulty, we can resort to local variation methods, where we replace the log-partition function with a suitable upper bound to obtain a tractable approximation. Upper bounds for log-partition functions are of great interest and are discussed in a number of papers (Wainwright et al., 2005; El Ghaoui and Gueye, 2008). A more recent approach taken by Khan et al. (2010) is to use a 'Bohning bound' on the log-sum-exponential function. This results in a tractable bound for categorical and binary variables, whose log-partition functions are more difficult to compute expectations with; other data types have easier log-partition functions whose bounds can be obtained using Jensen's inequality. This results in a variational method whose performance is shown to have comparable accuracy to the HMC approach we described here, but can be much faster.

A second approximate inference method is the *Integrated Nested Laplace Approximation* (INLA) (Rue et al., 2009), which allows for fast approximate inference in latent Gaussian models, and is thus appealing for the models we have discussed in this chapter. The INLA approach assumes that we have non-Gaussian observations $\mathbf{x}$, and latent variables $\mathbf{v}$ that are Gaussian and controlled only by a few hyperpa-

rameters $\boldsymbol{\vartheta}$.

INLA uses approximations to the marginal posterior density for the hyperparameters $\tilde{p}(\boldsymbol{\vartheta}|\mathbf{x})$, and for the full conditional marginal posterior densities $\tilde{p}(v_n|\mathbf{x}, \boldsymbol{\vartheta})$. The approximation for $p(\boldsymbol{\vartheta}|\mathbf{x})$ is given by a Laplace approximation, while the approximation for $p(v_n|\mathbf{x}, \boldsymbol{\vartheta})$ can be a Laplace or simplified Laplace approximation. The posterior marginals can then be computed using numerical integration:

$$\tilde{p}(v_n|\mathbf{x}) = \int \tilde{p}(v_n|\mathbf{x}, \boldsymbol{\vartheta})\tilde{p}(\boldsymbol{\vartheta}|x)d\boldsymbol{\vartheta} = \sum_{k=1}^{K} \tilde{p}(v_n|\mathbf{x}, \boldsymbol{\vartheta}_k)\tilde{p}(\boldsymbol{\vartheta}_k|x)\Delta_k, \tag{2.45}$$

where the area weights Δ_k are chosen either by using a grid of points or by the 'central composite design' (CCD) strategy, both of which are described by Rue et al. (2009) in detail. INLA has already been applied successfully in a number of settings (Rue et al., 2009; Martino et al., 2010; Yoon et al., 2010), and is an appealing approach for use in our model. The major limitation is that INLA requires a small number of hyperparameters to be effective, due to the numerical integration step. Some work already exists in overcoming this limitation (Yoon et al., 2010) making the use of INLA with the models we have described an interesting line of future work.

2.8 Latent Variable Models in Context

The development of latent factor models has a history over a century long with a specification in diverse areas of research including linear algebra, statistics, psychometrics, machine learning, biostatistics and computer vision, amongst others. The emergence of latent factor modelling can be traced to the method of Singular Value Decomposition (SVD) and its two progenitors, Eugenio Beltrami (1873) and Camille Jordan (1874) (Steward, 1993). These authors developed the ideas for SVD as part of a wider agenda for promoting an understanding of the class of bi-linear models. Of course, today this class of models is widely known, much used, and encompasses many of the models for matrix factorisation and latent variable modelling that have been developed since.

The model that has been the focus of much of this chapter, Principal Components Analysis (PCA) was initially specified by Pearson (1901) as a method for searching for the closest fitting lines and planes to points in space. At the same time, Factor Analysis (FA) was proposed by Spearman (1904) as a means of extracting factors of intelligence - much in the way factor analysis is used at present, though with the less lofty goal of explaining all human intelligence with the use of such methods. Both these methods are now part of the foundation of the modern study of unsupervised models with latent variables.

The modern development of this area of research begins with a move away from a linear-algebraic view, towards probabilistic interpretations of SVD and PCA, wherein the work of this chapter contributes. Machine learning research has been prolific in this area, with models focussing on the analysis of specific data types being actively developed. For *Gaussian data*, Tipping and Bishop (1997) and Roweis (1998) provided a a probabilistic interpretation of PCA by providing a generative model for SVD with a Bayesian analysis for PCA given by Bishop (1999). A focus on *binary data* led Tipping (1999) to propose a method for binary data visualisation using latent variable modelling and variational inference techniques, with Schein et al. (2003) specifying a similar logistic PCA. For *non-negative data*, the highly popular non-negative matrix factorisation (Paatero and Tapper, 1994; Lee and Seung, 1999) was presented with numerous non-negative variants of other methods being developed subsequently. For *co-occurrence* and multinomial data such as word appearances in documents, PLSA was developed (Hofmann, 1999) as well as its successor, Latent Dirichlet Allocation (LDA) (Blei et al., 2003). The relationship between these methods and the generalised models of this chapter were examined in section 2.2.2.

The unity of these distinct but related models was recognised by a number of authors. In psychometrics, Moustaki and Knott (2000) presented a generalised model for latent traits that considered an exponential family generalisation of models with latent variables – with the phrasing 'latent trait' being the term used in the psychometrics literature. An expectation maximisation based learning algorithm was described, but the model had problems with the numerical integration required and was demonstrated for two factors only. In machine learning, Collins et al. (2002), unaware of the work of Moustaki and Knott, proposed a generalised model for PCA. Collins et al. proposed a generic algorithm for parameter learning based on the alternating minimisation that was described in section 2.2.3.

Welling et al. (2008) followed by providing insights into the limits of maximum likelihood learning in the latent variable model framework discussing deterministic latent variable models, and proposed alternative inference based on variational methods. A family of probabilistic algorithms, called Discrete Components Analysis (DCA) was presented by Buntine and Jakulin (2006), and provided a unification of existing theory relating to latent variable models and dimensionality reduction with discrete distributions. The learning algorithms of the DCA family employ either Gibbs sampling or variational approximations. We developed fully Bayesian inference for generalised latent variable models in Mohamed et al. (2009) and have expanded on this work significantly in this chapter. In this chapter we go further by providing insight into the links between different models using exponential families, substitute link functions and Bregman divergences and expanding on the discussion on identifiability.

The literature on matrix factorisation is vast and a number of other papers are of relevance. Maximum Margin matrix factorisation (MMMF) was introduced by Srebro et al. (2005b) and bounds the norms of the matrix factors rather than the dimensionality and allows for an unbounded number of factors. Robust probabilistic projections (Archambeau et al., 2006) considers a Student's-t likelihood to handle data with outliers. Probabilistic matrix factorisation (Salakhutdinov and Mnih, 2008) was developed for the task of collaborative filtering and uses a Gaussian noise model for movie ratings data, and was shown to be scalable to large Netflix data set.

Additional generalised latent variable models of interest include: EPCA for belief compression in POMDPs (Roy and Gordon, 2003); models for supervised EPCA (Guo, 2008); sparse coding using EPCA (Lee et al., 2009); dynamic exponential family matrix factorisation (Hayashi et al., 2009); generalised models for spatially correlated multivariate data (Zhu et al., 2005); the Bayesian partial membership model (Heller et al., 2008); and Bayesian models for generalised spatial dynamic factor learning (Lopes et al., 2010).

2.9 Summary

In this chapter we developed a framework for generalising latent variable models to the exponential family. This exponential family generalisation extends the scope of latent variable models to data that is binary, categorical, counts, non-negative, or a heterogeneous set of these data types, and has unified many existing models. We have focussed on the promotion of Bayesian approaches to learning, after contemplating the limitations of the maximum likelihood approach. The mixture nature of the resulting generalised model and its identifiability properties were used to specify sampling schemes for learning and selecting a final embedding. We showed that the Bayesian approach is robust, avoids overfitting and is able to produce useful predictions in a number of settings.

Future research directions have already been alluded to by recent work, focussing on more complex data modalities such a spatial and time varying data. In what will prove to be a recurring observation, the ideas of this chapter pave the way for a parallel study of non-parametric Bayesian approaches to generalised modelling. Any future work, will at its core, become a study of the selection of prior distributions used in model construction. The next chapter takes one path in this line of thinking, by examining the implication of alternative priors for the latent variables. In particular, sparse priors will be examined, and will provide a new research direction where the generalised modelling framework will prove valuable.

Chapter 3

Models for Sparse Latent Factor Discovery

In this chapter we focus on *sparse latent representations*. A model is considered to be sparse if it sets to zero or close to zero any parameters that are not needed to explain the observed data. Sparsity allows the learning of parsimonious models that are interpretable and have gained in popularity, being well motivated in a number of application areas. We attempt to navigate the dichotomies that permeate current thinking in sparse learning: zero or close to zero, optimisation or Bayesian, shrinkage or discrete mixture priors, hypothesis or assumption. These issues are addressed using the framework for generalised learning developed in the previous chapter: by unifying models for sparse optimisation, designing new Bayesian models with sparsity and comparing these various approaches in a controlled manner.

3.1 Applications Motivating Sparse Representations

The analysis of data in any applied science comes with a wealth of domain knowledge that can be incorporated into the model building process. One property shared by data across scientific disciplines is an inherent redundancy in the data that allows for a sparse representation in some domain. Exploiting this sparsity can result in more effective model building and enhanced interpretability of model parameters. Three scientific areas where sparsity can be used to positive effect are used here to motivate an interest in methods for sparse learning.

One of the most prolific areas of research in sparse modelling is *computational biology*, where numerous motivating applications can be found. One common example where a sparse representation is applicable is in the analysis of gene expression data (Ishwaran and Rao, 2003; Huang et al., 2008; Carvalho et al., 2008).

A gene's expression is influenced by the presence of a number of transcription factor proteins, and there exists a wide array of such transcription factors that may affect the expression of any set of genes. Here, the underlying biology is considered to be sparse, since an individual gene's activity may only be directly influenced by a subset of the underlying transcription factors.

The *hedging problem* experienced in the construction of asset portfolios is a further area of interest (Brodie et al., 2009; Carvalho et al., 2010b). The financial market consists of many potential assets that can be used in hedging the risk of a portfolio. The high costs associated with creating a hedging portfolio with a large number of assets must be avoided to be profitable, requiring that only a subset of the available assets be used. Using sparse methods, the selection of an optimal subset of assets for hedging can be achieved and has the much sought after benefit of reducing transactional costs. The potential for effective hedging at lower cost and the concomitant prospect of higher profit provides a compelling motivation for the investigation of sparse methods in the construction of financial portfolios.

In *pharmacovigilance*, statistical analysis of adverse drug reactions (ADR) reported by patients is used in the surveillance of pharmaceutical products. The aim of pharmacovigilance is to highlight drugs that may cause adverse patient reactions (Caster et al., 2008; Madigan et al., 2010). Recent developments in the analysis of such data have moved away from pairwise evaluation of drugs when analysing adverse effects, to the use of multi-drug analysis methods. In a regression setting, 'interestingness' coefficients for problematic drugs are determined. This interestingness is used in the subsequent monitoring of any highlighted drugs and if ultimately necessary, provides a mechanism with which to accelerate the process of recalling harmful drugs. The sparse estimation of these coefficients is desirable since it makes interpretation easier by preventing confounding from other drugs appearing to be of interest. Due to the potential impact that drugs with adverse effects can have on the population, methods which improve this surveillance and ultimate recall are highly desirable, providing a strong motivation for the study of sparse methods in this setting.

Whether for methodological or application development, sparsity has come to play a prominent role in many settings, including statistical problems in normal-means estimation, regression, variable selection and dimensionality reduction, and applications in signal and image processing, compressed sensing and source separation. For unsupervised latent variable modelling - the focus of this thesis - models have been developed for sparse PCA (Zou et al., 2004; Zass and Shashua, 2006), sparse matrix factorisation (Srebro and Jaakkola, 2001; Dueck and Frey, 2004) and sparse factor regression models (Carvalho et al., 2008). This chapter will use

the tools developed in the previous chapter with a new focus on the role of prior specification for sparse learning in generalised latent variable models.

At the same time, the motivation for sparse methods does not come without a critique of its sensibilities. Do domain experts truly consider the gene regulation problems considered to be sparse (with exact zeroes)? Concurrently, sparsity in drug surveillance provides a compelling application of such methods. In all cases, whether one considers sparsity to be truly present or as an alternative methodology by which to explore data, it can be useful to agree that there is at least an underlying *compressibility* in most data sets that can be exploited to positive effect. Notwithstanding the philosophical aspects of these arguments, the remainder of this chapter will provide an exposition of current thinking in sparse and unsupervised learning. The more subtle aspects of learning with sparsity are probed in section 3.6.

3.2 Sparsity Inducing Loss Functions

An optimisation approach to sparse learning forms an intuitive basis upon which to consider the adaptation of existing methods. Such an optimisation strategy is based on the specification of a penalised loss function, using penalty functions that are known to encourage sparsity. Loss functions obtained in this manner often require different optimisation methods than those for unmodified loss functions and the development of these optimisation algorithms forms an active area of research.

3.2.1 L_p norm minimisation

A general penalised loss function based on the L_p norm has the following form:

$$\min_{\boldsymbol{\phi}} \sum_n \ell(\mathbf{x}_n, \boldsymbol{\phi}) + \alpha \|\boldsymbol{\phi}\|_p, \tag{3.1}$$

for any loss function of interest $\ell(\cdot)$, a D-dimensional data vector $\mathbf{x_n}$, model parameters $\boldsymbol{\phi}$, a regularisation parameter α, and the L_p norm $\|\cdot\|_p$ for $p \geq 0$. The L_p norm is defined as follows:

$$\|\mathbf{x}\|_0 = \sum_d \mathbb{I}(x_d \neq 0); \qquad \|\mathbf{x}\|_p = \left(\sum_d |x_d|^p \right)^{1/p}, \; p > 0. \tag{3.2}$$

If a loss function for regression is considered with $p = 2$, the familiar ridge regression is obtained. The use of the L_1 norm as a penalty function is well known to encourage sparse solutions, and was popularised by a model for sparse regression known as the LASSO (Tibshirani, 1996). Sparse solutions can also be obtained for the case of

$0 < p < 1$, and are briefly discussed in section 3.6.1.

An ideal approach to sparse learning would be to penalise parameters based on the number of non-zero elements, which can be achieved using the L_0 (quasi-) norm. This however, is an intractable combinatorial problem, requiring the enumeration of all subsets of sparse parameters and is thus not computationally feasible (Donoho, 2004). The majority of approaches to sparse learning in optimisation focus on L_1 norm penalisation. This popularity stems from an important result, often referred to as the $L_0 - L_1$ equivalence, that roughly states that if the representation to be computed is sufficiently sparse, then the NP-hard problem of finding the sparsest solution can be solved efficiently and exactly by minimizing an appropriate L_1 norm (Donoho, 2004, 2006). The convex nature of the L_1 norm has encouraged much development in optimisation strategies for L_1 norm minimisation, relying on the wide array of tools available from the theory of convex optimisation. The popularity of the L_1 norm has been further cemented by the rise in popularity of methods such as the LASSO (Tibshirani, 1996) and compressed sensing (Candes et al., 2006; Donoho, 2006).

3.2.2 Exponential Family PCA with Sparsity

We extend the exponential family PCA model discussed in section 2.2.2 using the sparse optimisation methodology described above using the L_1 norm. The resultant training objective for a sparse generalised latent variable model is:

$$\min_{V,\Theta} \sum_n \ell\left(\mathbf{x}_n, \mathbf{\Theta}\mathbf{v}_n\right) + \alpha\|\mathbf{V}\|_1 + \beta R(\mathbf{\Theta}), \tag{3.3}$$

where the loss function $\ell\left(\mathbf{x}_n, \mathbf{\Theta}\mathbf{v}_n\right) = -\ln p(\mathbf{x}_n|\mathbf{\Theta}\mathbf{v}_n)$ is the negative log likelihood function obtained using equation (2.28). The regularisation parameters α and β, control the degree to which the parameters $\mathbf{V}$ and $\mathbf{\Theta}$ are penalised and the function $R(\mathbf{\Theta})$ is any suitable regularisation function for the model parameters $\mathbf{\Theta}$. Importantly, equation (3.3) provides a unifying framework for sparse models with L_1 regularisation. This objective function is specified generally and is applicable for a wide choice of regularisation functions $R(\cdot)$, including the L_1 norm. Two loss function that can be obtained based on the choice of $R(\mathbf{\Theta})$ are:

Sparse MAP Loss. We use the loss function in equation (3.3) with $R(\mathbf{\Theta}) = -\ln p(\mathbf{\Theta}|\boldsymbol{\lambda}, \nu)$, which makes use of the conjugate prior distribution specified by equation (2.29). This corresponds to finding the maximum a posteriori (MAP) solution. This model will be referred to as sparse EPCA (SEPCA).

Sparse Parameter Loss. In addition to sparsity in $\mathbf{V}$, it is possible to include sparsity in $\mathbf{\Theta}$ using the L_1 norm using $R(\mathbf{\Theta}) = \|\mathbf{\Theta}\|_1$. While this may be an interesting model, its behaviour will not be considered further here.

The objective function for both of these functions is convex in either of its arguments with the other fixed, but is not convex in both arguments jointly. Similarly to the optimisation described in 2.2.3, we use an alternating minimisation procedure, which iteratively solves the following pair of optimisation problems:

$$\min_{\mathbf{V}} -\ln p(\mathbf{X}|\mathbf{V}, \mathbf{\Theta}) + \alpha\|\mathbf{V}\|_1 \tag{3.4}$$

$$\min_{\mathbf{\Theta}} -\ln p(\mathbf{X}|\mathbf{V}, \mathbf{\Theta}) + \beta R(\mathbf{\Theta}). \tag{3.5}$$

Since each individual optimisation remains convex, the extensive literature regarding L_1 norm regularisation can be referred to in solving these problems. The optimisation of equation (3.4) has been solved for the case of the Gaussian likelihood using the methods presented by Tibshirani (1996) in the LASSO. If a Bernoulli likelihood is considered, the optimisation corresponds to an instance of the L_1 regularised logistic regression (Lee et al., 2006b; Schmidt et al., 2007). For the general setting, a number of methods exist for solving this problem: it can be recast as an equivalent inequality constrained optimisation problem and solved using a modified LARS algorithm (Lee et al., 2006b), recast as a second order cone program or solved using a number of smooth approximations to the regularisation term (Schmidt et al., 2007), amongst others. The L_1 projection method of Schmidt et al. (2007) is used here and can be used in conjunction with any of the loss functions under study. Specific details of the optimisation scheme are deferred to that work.

3.3 Sparse Bayesian Learning

As opposed to the optimisation framework considered in the previous section, where one searches for the single best model parameters and variables, the Bayesian framework averages the model parameters and variables according to their posterior probability distribution, given the observed data. In the Bayesian setting, learning with sparsity involves the use of prior distributions that encourage sparsity. Prior distributions suitable for the purpose of sparse learning are referred to as sparsity-favouring priors. A sparsity-favouring prior can be any distribution centred at zero with high excess kurtosis, indicating that it is highly peaked with heavy tails or a distribution with a delta-mass at zero. The set of sparsity-favouring priors includes distributions such as the Normal-Gamma, Laplace (or double exponential) or Exponential distributions. Furthermore, distributions such as the Horseshoe (Carvalho et al., 2010a) or the spike-and-slab (Ishwaran and Rao, 2005) are suitable as sparse priors.

Table 3.1: Mixing densities used in the scale-mixture construction of various sparse priors.

Sparse Prior	**Mixing Density** $\pi(\lambda)$	
Student's-t	Inverse Gamma	$\mathcal{G}^{-1}\left(\lambda\vert\frac{\nu}{2},\frac{\nu}{2}\right)$
Laplace	Exponential	$\mathcal{E}(\lambda\vert\frac{1}{\nu})$
Normal/Jeffrey's	Reciprocal	$1/\lambda$
Horseshoe	Inverted Beta	$B'(\lambda\vert\frac{1}{2},\frac{1}{2})$
Normal-Gamma	Gamma	$\mathcal{G}(\lambda\vert\alpha,\frac{\beta^2}{2})$
Normal/Inverse-Gaussian	Inverse-Gaussian	$i\mathcal{N}(\lambda\vert\alpha,\beta)$
Normal/Exponential-Gamma	Exponential-Gamma	$(1+\lambda)^{-(c-1)}$

3.3.1 Continuous Sparsity Favouring Priors

There are numerous *continuous prior distributions* that have been used to encourage sparsity in the statistical literature. In most cases, these distributions share the property that they can be viewed as scale mixtures of Gaussian distributions (Andrews and Mallows, 1974; West, 1987). The scale-mixture of Gaussians is expressed by the following hierarchical specification for observed data $\mathbf{x}$ (Choy and Chan, 2008):

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\sigma}^2,\boldsymbol{\lambda}) = \prod_d p(x_d|\mu_d,\sigma_d^2,\lambda_d) \tag{3.6}$$

$$x_d|\mu_d,\sigma_d^2,\lambda_d \sim \mathcal{N}\left(x_d|\mu_d,\kappa(\lambda_d)\sigma_d^2\right) \tag{3.7}$$

$$\lambda_d \sim \pi(\lambda_d), \tag{3.8}$$

where $\kappa(\lambda_d)$ is a positive function of mixing parameters and $\pi(\lambda_d)$ is the mixing density on $\mathbb{R}^+$. λ_d is referred to as the global variance component and σ_d^2 as the local variance component. The scale mixture implies the following marginalisation:

$$x_d|\mu_d,\sigma_d^2 \sim \int_0^\infty \mathcal{N}\left(x_d|\mu_d,\kappa(\lambda_d)\sigma_d^2\right)\pi(\lambda_d)d\lambda_d. \tag{3.9}$$

For the implied marginal density to be suitable as a sparse prior, it must be shown that the resulting priors are peaked at zero and have tails that decay at a polynomial rate (i.e. decay according to some power law). A multitude of options for the mixing density are available that meet these requirements and yield priors suitable for sparse learning. Table 3.1 lists various sparse priors that can be obtained, assuming $\kappa(\lambda_d) = \lambda_d$ and using the listed mixing density. Contours of constant value are also shown for some commonly used sparse priors in figure 3.1.

Example 3.1: ***Normal-Gamma Distribution***

Consider the Normal-Gamma scale-mixture distribution:

$$p(x) = \int \mathcal{N}(x|0,\lambda)\mathcal{G}\left(\lambda|\alpha,\tfrac{\beta^2}{2}\right)d\lambda, \tag{3.10}$$

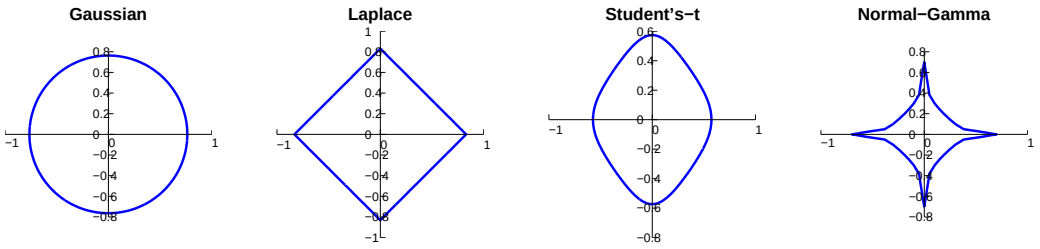


Figure 3.1: Contours of penalty functions associated with several sparse priors.

where the Gamma density $\mathcal{G}\left(\lambda|\alpha, \frac{\beta^2}{2}\right) = \frac{(\beta^2/2)^{\alpha/2}}{\Gamma(\alpha/2)}(\lambda)^{\alpha-1}\exp\left(\frac{-\beta^2}{2}\lambda\right)$, with $\alpha, \beta > 0$ are known constants. The marginal pdf for $x \neq 0$ is:

$$p(x) = \frac{\beta^{\alpha+\frac{1}{2}}}{\sqrt{\pi}2^{\alpha-\frac{1}{2}}\Gamma(\alpha)}|x|^{\alpha-\frac{1}{2}}\mathcal{K}_{\alpha-\frac{1}{2}}(\beta|x|), \tag{3.11}$$

where $\mathcal{K}_{\alpha-\frac{1}{2}}(\cdot)$ is the modified Bessel function of the second kind. This density is a member of the class of generalised hyperbolic distributions, which includes other distributions such as the Normal-Inverse Gaussian (Barndorff-Nielsen, 1978). The sparsity-favouring properties of the Normal-Gamma distribution can be evaluated by examining its properties at zero and the tail behaviour of equation (3.11):

$$\lim_{x\to 0} p(x) = \begin{cases} \frac{\beta}{2\sqrt{\pi}}\frac{\Gamma(\alpha-\frac{1}{2})}{\Gamma(\alpha)} & \text{for } \alpha > \frac{1}{2} \\ \infty & \text{otherwise} \end{cases} \tag{3.12}$$

$$p(x) \propto |x|^{\alpha-1}\exp(-x) \text{ for } x \gg \left|\left(\alpha-\tfrac{1}{2}\right)^2 - \tfrac{1}{4}\right|, \tag{3.13}$$

where the above two equations can be derived by using the asymptotic forms of the Bessel function (NIST, 2010, eq. 10.30, 10.41). These two properties show that the density is highly peaked at zero and has tails with polynomial decay, and is thus suitable as a sparse prior. □

A characteristic of these priors is that these continuous densities place no mass on zero itself and the samples never contain exact zeroes. If we believe that that the latent representation should contain exact zeroes, then a prior with a delta mass at zero must be used.

3.3.2 Sparsity with Spike-and-Slab Priors

The second class of sparse priors that can be used are based on a discrete mixture of point mass at zero, referred to as the 'spike' and any other distribution known as the 'slab', giving the alternative name as a 'spike-and-slab' distribution (Mitchell and Beauchamp, 1988; Ishwaran and Rao, 2005). Sparsity in the latent variables v_{nk}, for $\mathbf{v}_n = [v_{n1}, \ldots, v_{nK}]$, is encoded by considering independent prior distributions given

by the mixture:

$$p(\mathbf{v}_n|\mathbf{z}_n) = \prod_k p(v_{nk}|z_{nk}) \tag{3.14}$$

$$p(v_{nk}|z_{nk}) = (1 - z_{nk})\delta_0(v_{nk}) + z_{nk}\pi(v_{nk}), \tag{3.15}$$

where δ_0 is the delta function at zero and $\pi(v_{nk})$ is assumed to be a fixed unimodal symmetric density, often a uniform or Gaussian distribution. Since this prior places mass explicitly on zero, it is suitable as a sparse prior resulting in Bayesian inference with exact zeroes in any samples obtained. The spike-and-slab distribution has enjoyed application in a wide range of statistical problems including regression and variable selection (Ishwaran and Rao, 2005; O'Hara and Sillanpäa, 2009).

For the practical use of this prior, we construct the the spike-and-slab using a K-dimensional binary vector $\mathbf{z}_n$, which indicates whether an individual parameter v_{nk} is sampled with probability π_k from the slab component or if it is to be sampled from the spike. We use a hierarchical specification with Bernoulli indicator variables and Beta priors for the spike/slab probability π_k.

$$p(\mathbf{z}_n|\boldsymbol{\pi}) = \prod_k \mathcal{B}(z_{nk}|\pi_k) = \prod_k {\pi_k}^{z_{nk}}(1 - \pi_k)^{1-z_{nk}} \tag{3.16}$$

$$p(\pi_k|e, f) = \beta(\pi_k|e, f) = \frac{1}{B(e, f)}{\pi_k}^{e-1}(1 - \pi_k)^{f-1}. \tag{3.17}$$

The Beta function is $B(e, f) = {\Gamma(e + f)}/{(\Gamma(e)\Gamma(f))}$. For the choice of a Gaussian slab, the spike decisions are combined with the slab to form the overall probability:

$$p(\mathbf{v}_n|\mathbf{z}_n, \mathbf{m}, \boldsymbol{\Sigma}) = \prod_k \mathcal{N}(v_{nk}|z_{nk}m_k, z_{nk}\sigma_k^2), \tag{3.18}$$

where the mean of the Gaussian is m_k and the diagonal covariance $\boldsymbol{\Sigma}$ has elements σ_k^2. For this definition, when $z_{nk} = 0$, $p(v_{nk})$ in equation (3.18) becomes a delta function at zero, indicating that the spike has been chosen instead of the slab.

This construction is particularly interesting, since it can be interpreted as a penalty on the number of non-zero elements, in the same manner that the L_0 norm would penalise model parameters. The expected L_0 norm of $\mathbf{v}$ can be computed as:

$$card(\mathbf{v}_n) = \mathbb{E}\left[\|\mathbf{v}_n\|_0\right] = \mathbb{E}\left[\sum_{k=1}^{K} z_{nk}\right] = \sum_{k=1}^{K} \mathbb{E}\left[z_{nk}\right] = K\frac{e}{e + f}, \tag{3.19}$$

where $1 \leq card(\mathbf{v}_n) \leq K - 1$ for sparse representations of $\mathbf{v}_n$. Under suitable scaling of the hyperparameters: $e \to {e}/{K}$ and $f \to f \cdot ({K - 1}/{K})$, the cardinality $card(\mathbf{v}_n) \sim \mathcal{P}({e}/{f})$ as $K \to \infty$. This is obtained by recalling that in the limit, the binomial distribution can be approximated by a Poisson distribution. This analy-

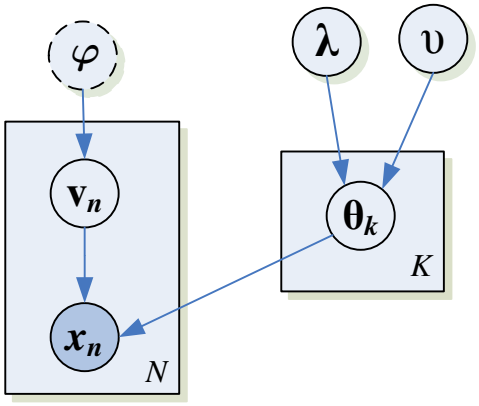


Figure 3.2: Generic graphical model for learning in latent variable models with sparsity.

sis gives insight into the behaviour of the prior as well as some guidance in setting hyperparameter values.

3.3.3 Learning in Latent Variable Models with Sparsity

The classes of priors discussed in the previous two sections are easily incorporated into the framework for generalised latent variable models. The modelling will focus on the incorporation sparsity in the latent variable $\mathbf{V}$ only.

Figure 3.2 shows a generic form of the graphical model described in section 2.3.2, and is given again here for clarity. The plate notation represents replication of variables and the dashed node $\boldsymbol{\varphi}$ represents any appropriate hyper-prior distribution for the latent variables $\mathbf{v}_n$. The observed data forms a $D \times N$ matrix $\mathbf{X}$, with columns $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$. N is the number of data points and D is the number of observed dimensions. $\boldsymbol{\Theta}$ is a $D \times K$ matrix with rows $\boldsymbol{\theta}_k$. $\mathbf{V}$ is an $K \times N$ matrix $\mathbf{V}$, with columns $\mathbf{v}_n = [v_{n1}, \ldots, v_{nK}]$, where K is the number of latent factors.

The required conditional distributions are:

$$\mathbf{x}_n|\mathbf{v}_n, \boldsymbol{\Theta} \sim \text{Expon}\left(\sum_k v_{nk}\boldsymbol{\theta}_k\right) \tag{3.20}$$

$$\boldsymbol{\theta}_k \sim \text{Conj}\left(\boldsymbol{\lambda}, \nu\right). \tag{3.21}$$

The joint probability is thus:

$$p(\mathbf{X}, \boldsymbol{\Omega}|\boldsymbol{\Psi}) = p(\mathbf{X}|\mathbf{V}, \boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\lambda}, \nu)p(\mathbf{V}|\boldsymbol{\varphi}), \tag{3.22}$$

where $\boldsymbol{\Omega}$ is the set of unknowns to be learnt and $\boldsymbol{\Psi}$ is the set of model hyperparameters. The model specification is completed by the choice of sparse prior for

the latent variables $\mathbf{V}$, of which the two classes of priors discussed will be considered separately here. The nature of these two classes require different approaches to learning and here we focus solely on Markov chain Monte Carlo (MCMC) methods for learning.

3.3.3.1 Learning with Continuous Priors

We consider the following candidate models:

Laplace Model. We use the Laplace or double exponential prior:

$$\mathbf{v}_n \sim \prod_{k=1}^{K} \mathcal{L}(v_{nk}|b_k) \quad = \quad \prod_{k=1}^{K} \tfrac{1}{2} b_k \exp\left(-b_k|v_{nk}|\right). \tag{3.23}$$

This choice of model allows for a Bayesian analogue of the sparse EPCA model described in section 3.2.2. This model will be referred to LXPCA. The equivalence between this model and the sparse EPCA model described previously, can be seen by comparing the log-joint probability probability using the Laplace prior in equation (3.22) to the sparse MAP loss described for equation (3.3).

Exponential Model. We also use the exponential distribution:

$$\mathbf{v}_n \sim \prod_{k=1}^{K} \mathcal{E}(v_{nk}|b_k) \quad = \quad \prod_{k=1}^{K} b_k \exp\left(-b_k v_{nk}\right). \tag{3.24}$$

This distribution has similar shrinkage properties to the Laplace. In addition, since the distribution has support on the positive real line, it allows for non-negative representations of the latent space, such that $v_{nk} \geq 0$. This model will be referred to as NXPCA.

The above two model types have been considered for the case of sparse generalised linear models for regression by Seeger et al. (2007). The hierarchical specification is completed by placing a Gamma prior on the unknown rate parameters $\mathbf{b}$, with shared shape and scale parameters α and β respectively. The set of unknown variables to be inferred is denoted as $\mathbf{\Omega} = \{\mathbf{V}, \mathbf{\Theta}, \mathbf{b}\}$ and the set of hyperparameters as $\mathbf{\Psi} = \{\alpha, \beta, \boldsymbol{\lambda}, \nu\}$.

The experience we have gained in developing the sampling scheme for the Bayesian exponential family PCA model (BXPCA) is used here. We use Hybrid Monte Carlo sampling, where the required potential energy function is: $\mathcal{E}(\mathbf{\Omega}|\mathbf{\Psi}) = -\ln p(\mathbf{X}, \mathbf{\Omega}|\mathbf{\Psi})$. Constrained parameters such as $b_k > 0$ in both models above, and $v_{nk} \geq 0$ in the exponential case are transformed to unconstrained parameters using the transformation $b_k = \exp(\xi_k)$ and $v_{nk} = \exp(\chi_{nk})$. The learning method is also adapted to handle missing data using the method described for BXPCA in section 2.3.4.

3.3.3.2 Learning with the Spike-and-Slab

For the discrete mixture prior using a Gaussian slab, we use the following model:

Spike-and-Slab Model. The prior distribution used is:

$$p(\mathbf{v}_n|\mathbf{z}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_k \mathcal{N}(v_{nk}|z_{nk}\mu_k, z_{nk}\sigma_k^2), \tag{3.25}$$

where the definition of $\mathbf{z}_n$ is given by equation (3.16) and the construction of the prior is described in section 3.3.2. The mean and variance of the Gaussian slab component are μ_k and σ_k^2 respectively. Here, the set of unknown variables to be inferred is $\boldsymbol{\Omega} = \{\mathbf{Z}, \mathbf{V}, \boldsymbol{\Theta}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and the set of hyperparameters $\boldsymbol{\Psi} = \{e, f, \boldsymbol{\lambda}, \nu\}$.

Since $\mathbf{Z}$ is discrete, the required sampling is more difficult. We develop a sampling approach using Metropolis-within-Gibbs sampling, where each of the unknown variables are sequentially sampled using Metropolis-Hastings. The sampling proceeds by iterating over the following steps:

1. Sample $\mathbf{Z}$ and $\mathbf{V}$ jointly using a pairwise sampling for the latent variable pair (z_{nk}, v_{nk}).
2. Sample $\boldsymbol{\Theta}$ by slice sampling.
3. Sample $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$ by Gibbs sampling.

Sampling Z and V. Sampling the latent factors z_{nk} and v_{nk}, involves the two step procedure of deciding whether a latent factor contributes to the data or not by sampling z_{nk} having integrated out v_{nk}. All variables v_{nk} associated with the slab components are sampled using slice sampling. The decision to choose either the spike or the slab involves the following probabilities:

$$p(z_{nk} = 0|\mathbf{X}, \boldsymbol{\pi}, \mathbf{V}_{\neg nk}) \quad \text{and} \quad p(z_{nk} = 1|\mathbf{X}, \boldsymbol{\pi}, \mathbf{V}_{\neg nk}), \tag{3.26}$$

where $\mathbf{V}_{\neg nk}$ are current values of $\mathbf{V}$, with v_{nk} excluded. Based on this decision, the latent variable is sampled from the spike or the slab component. Evaluating these probabilities involves computing the following integrals:

$$\begin{aligned} p(z_{nk} = 0|\mathbf{X}, \boldsymbol{\pi}, \mathbf{V}_{\neg nk}) &\propto \int p(z_{nk} = 0, v_{nk} = 0, \mathbf{X}|\mathbf{V}_{\neg nk}, \boldsymbol{\pi}) dv_{nk} \\ &= (1 - \pi_k) p(\mathbf{X}|\mathbf{V}_{\neg nk}, v_{nk} = 0, \boldsymbol{\Theta}). \end{aligned} \tag{3.27}$$

$$\begin{aligned} p(z_{nk} = 1|\mathbf{X}, \boldsymbol{\pi}, \mathbf{V}_{\neg nk}) &\propto \int p(z_{nk} = 1, v_{nk}, \mathbf{X}|\mathbf{V}_{\neg nk}, \boldsymbol{\pi}) dv_{nk} \\ &= \pi_k \int p(\mathbf{X}|\mathbf{V}_{\neg nk}, v_{nk}, \boldsymbol{\Theta}) \mathcal{N}(v_{nk}|\mu_k, \sigma_k^2) dv_{nk}. \end{aligned} \tag{3.28}$$

While computing (3.27) is easy, the integral in equation (3.28) is not tractable

in general. While it may be computed for certain exponential families such as the Gaussian, for other families the integral must be approximated. Any approximation method can be used, such as Monte Carlo Integration or the Laplace approximation. Laplace's method is used here (MacKay, 2003, ch. 27).

The use of the Laplace method introduces an error due to the approximation of the target distribution. This problem has been studied by Guihenneuc-Jouyaux and Rousseau (2005) where the Laplace approximation is used in MCMC schemes with latent variables such as in our case, and show that such an approach can behave well. Guihenneuc-Jouyaux and Rousseau (2005) show that as the number of observations increases, the approximate distribution becomes close to the true distribution, and describe a number of assumptions for this to hold, such as requiring differentiability, a positive definite information matrix and conditions on the behaviour of the prior at boundaries of the parameter space.

It is possible to avoid this approximation altogether by using the pseudo-marginals approach discussed by Andrieu and Roberts (2009), which is useful in MCMC settings where we have a term, say $p(z)$, that is difficult to compute, such as equation (3.28). The idea underpinning the pseudo-marginal approach is that if the difficult to compute term $p(z)$ can be replaced by an easier to compute unbiased estimator $r(z)$ in the Metropolis-Hastings acceptance ratio (e.g., by an importance sampling estimate as used by Beaumont (2003)), then the Markov chain will have an equilibrium distribution that is exactly $p(z)$. Andrieu and Roberts (2009) explain in detail the workings of this approach and the conditions for validity, making the pseudo-marginals method an appealing methods for improving this step of the sampling scheme.

Slice Sampling of $\boldsymbol{\Theta}$. Both $\mathbf{V}$ and $\boldsymbol{\Theta}$ can be sampled by slice sampling. The method of slice sampling, described in section 1.5.3, is a general version of the Gibbs sampler (Neal, 2003), and proceeds to sample all parameters in a co-ordinate-wise fashion. Sampling requires the evaluation of the joint-probability of all parameters of interest. To sample $\boldsymbol{\Theta}$, the required joint probability is:

$$\ln p(\mathbf{X}, \boldsymbol{\Theta}) \quad = \quad \ln p(\mathbf{X}|\mathbf{V}, \boldsymbol{\Theta}) + \ln p(\boldsymbol{\Theta}|\boldsymbol{\lambda}, \nu), \tag{3.29}$$

which can be easily evaluated. A similar evaluation is needed for $\mathbf{V}$.

Gibbs Sampling $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$. The variables $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and $\boldsymbol{\pi}$, have conjugate relationships with the latent variables $\mathbf{V}$ and $\mathbf{Z}$ respectively. Gibbs sampling is a natural choice since the full conditional distributions are easily derived. These full conditionals are omitted here for brevity (Gilks et al., 1995). The full conditional distributions that are required for $\boldsymbol{\pi}$ are derived in example 1.4.

3.3.4 Implications of Bayesian Learning with Sparsity

The two classes of priors introduced: the continuous sparsity-favouring and the discrete mixture priors, give rise to two notions of strong and weak sparsity.

Strong Sparsity. A vector $\boldsymbol{\omega}$ is considered to be 'strongly sparse' if elements of $\boldsymbol{\omega}$ are exactly zero. The spike-and-slab prior places mass explicitly on zero and is thus a prior suited to achieving this notion of sparsity in parameter learning. We can also think of this in terms of the structure of a graph, where this notion of sparsity expresses uncertainty in the connectivity structure of the graph.

Weak sparsity. A vector $\boldsymbol{\omega}$ is considered to be 'weakly sparse' if none of its elements are exactly zero, but which has a small number of elements with large entries, and other elements close to zero. This implies that a weakly sparse vector $\boldsymbol{\omega}$ has a small L_p norm for small p or has entries which decay in absolute value according to some power law (Johnstone and Silverman, 2004). When thinking of graph structure, this type of sparsity assumes that the structure of the graph is given and the uncertainty is in the strength of connections between nodes.

There remains no clear choice between using one type of sparsity over the other. Certain practitioners may implicitly refer to sparsity as a strong sparsity as a matter of definition. Using a representation with exact zeroes brings with it an easier interpretation of coefficients in the model, as well as computational advantages in terms of storing fewer elements in memory.

The rapid combinatorial growth of the solution set may be of concern when using discrete mixture priors. This is especially of concern in the 'large p' paradigm (West, 2003), particularly in applications concerned with the analysis of genomic data where the dimensionality (D as used here) of the data is very large. Methods for high-dimensional analysis in this setting were discussed by Carvalho et al. (2008). These methods encode sparsity in the factor loadings $\boldsymbol{\Theta}$, which scale with D and may become problematic when D is very large. In contrast, the models discussed in this chapter simulate sparsity in the latent factors $\mathbf{V}$, which scale with the number of latent factors K. Since $K \ll D, N$, the inference scheme presented here is less prone to problems in simulating the configuration of sparse elements.

Continuous sparsity-favouring priors never place any mass on zero itself, resulting in weak sparsity, with strong sparsity obtained only by thresholding. Practitioners may, for philosophical reasons, be averse to including exact zeroes in model parameters and find it preferable to consider the continuous sparsity-favouring case (Gelman et al., 2004, pp. 180). A further aspect of strong sparsity deals with model averaging. Any model averaged coefficients will be non-zero, even with the use

of discrete mixture priors, which makes the use of continuous priors seem favourable.

Both types of priors are immensely popular, having proven to be effective in a number of applied settings. In both cases, the prior aims to place substantial mass on or near zero, and to provide a mechanism by which model parameters that contribute to explaining the observed data are not shrunk towards zero. Continuous sparsity favouring priors enforce a global shrinkage on model parameters. It is this property that induces sparsity by shrinking parameter values towards zero, but which also results in shrinkage of parameters of relevance to the data. It is to accommodate these parameters of relevance, that the need for heavy tailed priors arises. Simultaneous global and local shrinkage is performed by the discrete mixture prior, which has the ability to give both sparsity in the model parameters, while not restricting the parameters that contribute to explaining the data. These operational differences are important and will be examined in the experimental analysis.

3.4 Comparing Model Performance

We use the testing methodology described in section 2.4.1 to evaluate the performance of the sparse models developed in this chapter. All the sparse methods discussed are tested using a test set consisting of 10% of the data elements. For fairness in evaluation, we choose the regularisation parameters α and β, described for SEPCA in section 3.2.2, by cross-validation using a validation data set chosen as 5% of the data elements. This validation set is independent of the data that has been set aside as training or testing data.

3.4.1 Analysis using Synthetic Data

As a synthetic benchmark data set, we use the block images data from Griffiths and Ghahramani (2006). The data consists of 100 6×6 binary images, with each image $\mathbf{x}_n$ represented as a 36-dimensional vector. We generated the images with four latent features, each being a specific type of block and the observed data is a combination of a number of these latent features. We flipped each bit in the resulting data set with a probability of 0.1, thus adding noise to each of the images. This data set is useful as a benchmark since it consists of a number of latent factors, but only a sparse subset of these factors may contribute to explaining any single data point. This data is synthetic but was not generated from any of the models tested. The four base images and representative training examples are shown in figure 3.3a.

Figure 3.3b shows the predictive probability (NLP) and root mean squared error (RMSE) on this benchmark data set. The sparse models we developed are

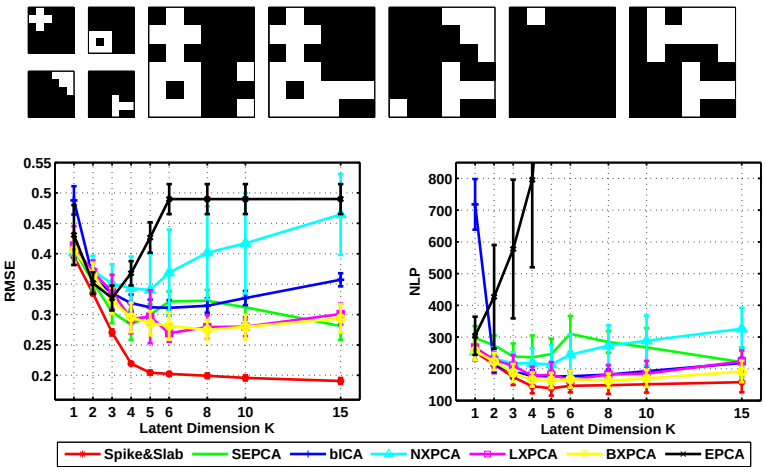


Figure 3.3: (a) *Row 1:* Samples of the training data used. The first panel block shows the base images used to construct the data. (b) *Row 2:* RMSE and NLP for various latent dimensions on the block images data set.

compared to EPCA (Collins et al., 2002), BXPCA (Mohamed et al., 2009, c.f. chapter 2 here) and to binary ICA (Kaban and Bingham, 2006). A random predictor would have an $NLP = 100 \times 36 \times 10\% = 360$ bits. The models tested here have performance significantly better than this. All models are able to find the appropriate number of latent dimensions as either four or five. Models that choose five latent factors tend to make specific allowances for a null factor, where none of the factors are combined to make an image. The behaviour of BXPCA and EPCA is consistent with the understanding of these models developed in the previous chapter. The spike-and-slab model shows the best performance with smaller error bars.

3.4.2 Application to Real World Data

Robot Planning. The robot planning data set of Kollar and Roy (2009) consists of tags of objects in $N = 750$ images taken by a robot-mounted camera in an office area. The tags were acquired by hand annotation and indicate whether objects such as bikes, computers screens or doors, appear in the images, with $D = 23$ of the most popular tags being used. Figure 3.4 shows the test RMSE for five latent dimensions for all the methods discussed in this chapter.

SPECT Images. Data of cardiac Single Proton Emission Computed Tomography (SPECT) images is used (UCI Data) and consists of $N = 267$ SPECT images that have been pre-processed resulting in $D = 22$ binary attributes. We present RMSE with five latent dimensions in figure 3.4.

Animal Descriptions. In a study by Kemp and Tenenbaum (2008), an adult participant was asked to make binary judgements as to which of a set of $D = 102$ characteristics applied to $N = 33$ animals. The animal characteristics that were evaluated included perceptual ('is black'), anatomical ('has feathers'), ecological

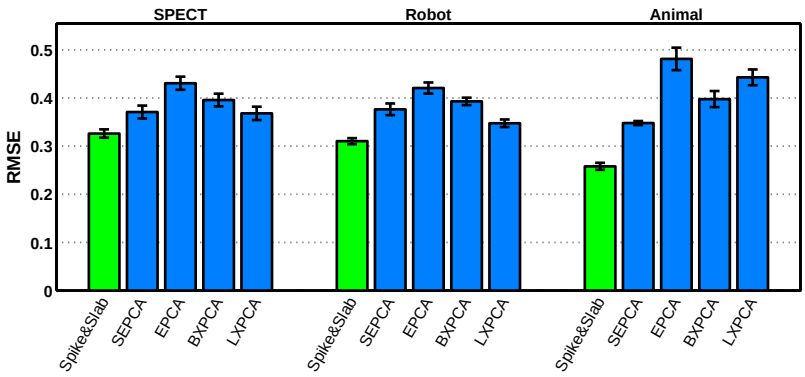


Figure 3.4: Comparisons of RMSE obtained for various sparse methods using three real world data sets for $K = 5$ latent dimensions.

('lives in a hole') and behavioural features ('travels in groups'). The RMSE on held-out data for five latent factors is shown in figure 3.4. This data set will be examined further in section 3.5.

In all three cases, the spike-and-slab has the best reconstruction performance on the held out data.

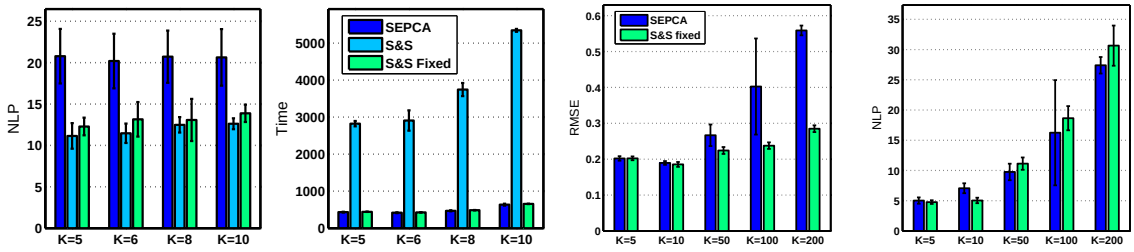
The '$p > n$' paradigm: The performance of the sparse methods presented are also discussed for the case where the observed dimensionality D is larger than the number of observations N.

Newsgroups Text. A subset of the popular 20 newsgroups data set was used (UCI Data), which consists of documents and counts of the words used in each document. We use $N = 100$ articles with $D = 200$ words, having a data sparsity of 93%. Here, the model uses a Possion likelihood to model the word counts. Figure 3.5a shows the performance of the spike-and-slab model and SEPCA. Apart from the application of the model to count-based data, the results show that the spike-and-slab model is able to deal effectively with the sparse data, and provides effective reconstructions and good predictive performance on held out data. S&S fixed in the figure 3.5a is the performance of the spike-and-slab when its running time is fixed to that taken for the optimisation of sparse EPCA and shows efficient performance in this setting. Table 3.2 shows that the number of non-zeroes in the reconstructions for various K, with the true number of non-zeroes being 1436. SEPCA is very poor at learning the structure of this sparse data set, whereas the spike-and-slab is robust to the data sparsity. This aspect will be discussed further in the ensuing discussion.

The common lore regarding computation time is that MCMC methods are dramatically slower than optimisation methods. In general, MCMC methods do not always scale poorly, even in comparison to optimisation methods, as demonstrated by Salakhutdinov and Mnih (2008) for example. The cross-validation procedure

Table 3.2: Number of non-zeroes in newsgroups data reconstruction for both SEPCA and S&S. The true number of non-zeroes is 1436.

K	5	6	8	10
SEPCA	475 ± 36	483 ± 57	592 ± 207	934 ± 440
Spike-Slab	1446 ± 24	1418 ± 29	1400 ± 18	1367 ± 32



(a) Newsgroups data (counts) (b) Hapmap data (binary)

Figure 3.5: Time matched performance analysis for: (a) newsgroups data using a Poisson likelihood, and (b) hapmap data using a Bernoulli likelihood. S&S fixed is the time matched spike-and-slab performance.

needed to set regularisation parameters α and β is computationally demanding due to the need to execute the optimisation for many combinations of parameters. This approach is also wasteful of data since a separate validation data set is needed to make sensible choices for these parameters and to avoid model overfitting. While individual optimisations may be quick, the overall procedure can take an extended time and depends on the granularity of the grid over which regularisation values are searched for. These parameters can be learnt in the Bayesian setting and have the advantage that we obtain information about the distribution of the parameters, rather than point estimates and can have greatly improved performance.

For the the newsgroups data, figure 3.5a demonstrates this trade-off between running time and performance of the optimisation and the Bayesian approaches. The comparison shows the running times of the spike-and-slab inference (S&S) for 200 iterations, and SEPCA run to convergence. The figure gives the impression that the Bayesian spike-and-slab is slower by a factor of 2.5 for this data set. But the performance when measured using predictive probability is dramatically better. We adjusted the testing methodology to consider the consider the setting where we fixed the running time for the spike-and-slab model – this running time being dictated by the running time of the SEPCA optimisation method. The results are shown as S&S fixed in figure 3.5a and show that even with a fixed time budget, MCMC performs better in this setting. The same result is shown for the hapmap data in figure 3.5b, with the Bayesian approach having a much lower NLP in the time matched case and with fixed computation budget.

Hapmap Data. The Hapmap data set consists of Single Nucleotide Polymorphisms (SNPs) that indicate DNA sequence variations between individuals in a population (Marchini et al., 2007). The data from $N = 100$ individuals using $D = 200$ positions is used. Figure 3.5b shows the performance of the spike-and-slab model and SEPCA, using the time-matched methodology just described. The spike-and-slab has similar reconstruction performance as SEPCA in terms of RMSE, at low latent dimensionality, but much better performance as number of latent factors K approaches the size of the data D. The graph of performance on predictive probability remains highly comparable to SEPCA but shows overlapping error bars as K becomes close to D, which suggests that without the time constraint further improvements can be made.

The spike-and-slab performs both a local and global shrinkage and has the ability to adapt to the global sparsity but assesses locally the importance of latent variables. Other priors such as the Laplace prior perform only a global shrinkage and must simultaneously learn the sparsity pattern and the contributions of the latent variables, which results in a tradeoff between the two with reduced performance. Similar observations, particularly for the case of the Laplace distribution, have been made by Scott and Berger (2006, pp. 156), noting that the Laplace lacks both enough mass near zero and tails that are sufficiently heavy for robust estimation.

3.5 Study: Discerning Mental Models of Animals

The data set of human judgements of animal characteristics was described in section 3.4.2. The study by Kemp and Tenenbaum (2008) aimed to gain insight into the mental models or structured forms used by humans in understanding related concepts. One means of understanding this is to infer the set of underlying factors that the human subject believes is shared by certain animals, but not by others. These underlying factors provide insight into the structure used in understanding the relationships between various animals, and that is an inherent part of the user's mental model of animals.

We use the spike-and-slab model to infer underlying factors for this data set. A visualisation of the latent embedding is useful in understanding the structural relationships involved. Our model with sparsity in the latent factors $\mathbf{V}$ is especially appropriate for this study because it aims to describes the relationship between the animals (observations) and the underlying factors. Figure 3.6 shows the 3-dimensional embedding of the animals obtained using a single sample from the Markov chain at convergence from the spike-and-slab model. The plot shows clear groupings of animals: insects (Butterfly, Bee) in the bottom left and a separation of terrestrial animals (Giraffe, Dog, Gorilla) from avian and aquatic animals (Whale,

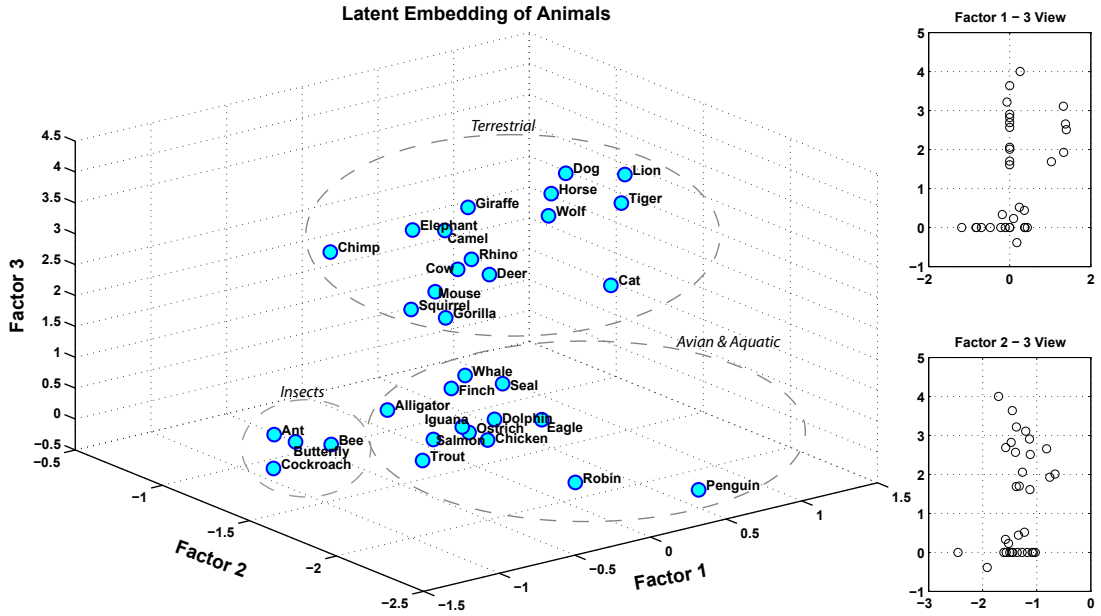


Figure 3.6: Visualisation of the animal embedding. The right-hand side plots show 2D perspectives of the factors to depict the sparsity pattern.

Chicken). These groupings match norms associated with an adult understanding of animals. These groupings are also similar to the structural forms discussed by Kemp and Tenenbaum (2008, fig. 5), and show that the underlying factors are able to provide a meaningful representation of the data.

Figure 3.7 shows results for various latent dimensions for NLP and RMSE, under the testing methodology used throughout this chapter. For this data, the NLP of a random classifier is 336 bits and the models have NLP values much lower than this. Factors between 4 and 10 are appropriate number of factors to explain the data. We applied the time-matched testing methodology to this data set as another test of the run-time behaviour of the Bayesian spike-and-slab method in relation to the optimisation-based approach. Figure 3.8 shows the running times for the two methods and shows that even with a fixed time budget determined by the optimisation based approach, the Bayesian method is able to produce improved reconstructions. A more elaborate analysis of such data would involve responses from multiple participants to investigate shared characteristics of mental models across the set of participants. For this setting, tensor models of the type discussed in chapter 5 would be appropriate.

3.6 Discussion

We examine some of the important considerations that arise from the approaches to sparse learning developed in this chapter. Here we look at ways of including further

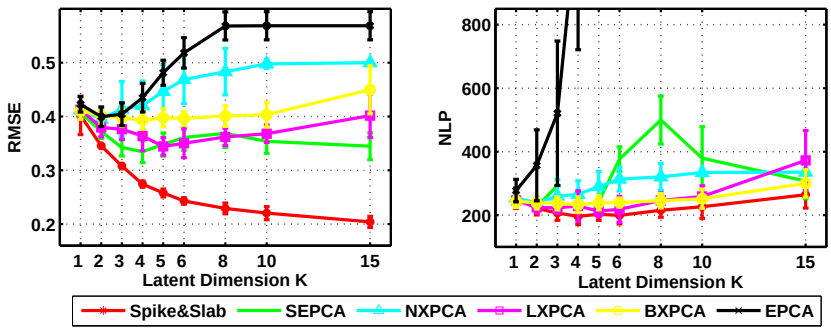


Figure 3.7: RMSE and NLP comparisons for the human judgements data for various latent dimensions K.

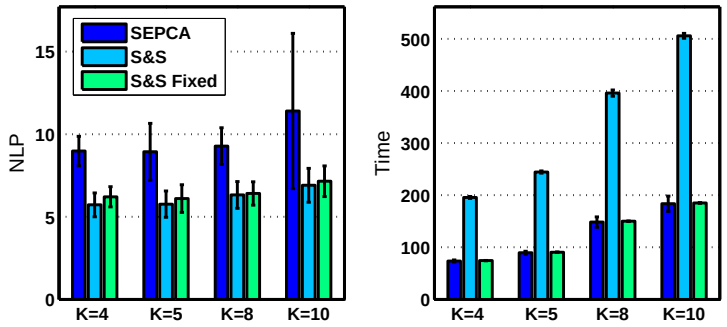


Figure 3.8: Timing analysis for the human judgements data set.

structure into the sparse representations and the penalty functions that are used. We also describe two related areas of research to which the sparse methods developed in this chapter have a strong connection and can be integrated into.

3.6.1 Beyond L_1 Penalisation

Other penalties beyond the L_1 norm exist, though they are less widely used. The use of these other penalties is motivated by the potential for discovering faster or more powerful algorithms for sparse learning. The non-negative Garrote (Breiman, 1995) is a method for variable selection that shrinks the least squares regression estimate by multiplying them by shrinking factors, whose sum is constrained, rather than the L_1 norm as used in the Lasso. The development of the Lasso was inspired by the non-negative Garrote, as a means of removing the reliance on the least squares estimate. The Lasso does not satisfy the oracle properties: identifying the correct subset of variables, and achieving the optimal estimation rate. The Lasso also produces biased estimates for large coefficients. To overcome these shortcomings, the adaptive Lasso (Zou, 2006) was designed, and achieves the oracle properties using a weighted L_1 penalty, employing different weights for each of the model coefficients; the weights being data dependent. van de Geer and Bühlmann (2009) provide a comprehensive review of the oracle results for the Lasso and the modifications and conditions required to achieve them.

One class of penalties beyond the L_1 norm, are compound norms which combine the L_1 norm with additional constraints. The general objective (3.1) forms the basis of these more complex objective functions. Examples that fall into this category include the L_1 optimisation over subsets of variables in either the group Lasso (Yuan and Lin, 2006) or the fused Lasso (Tibshirani et al., 2005) or the inclusion of smoothness properties using the total variation norm (Lustig et al., 2007). Consider a D-dimensional vector $\mathbf{x}$ of parameters, the vector of differences $\Delta\mathbf{x}$ with elements $(\Delta\mathbf{x})_i = x_i - x_{i-1}$. The fused Lasso penalty incorporates smoothness into the parameter estimation and has the following form:

$$\textbf{Fused Lasso:} \quad R(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + \beta\|\Delta\mathbf{x}\|_1, \tag{3.30}$$

where α and β are regularisation parameters. A similar penalty for groups of variables can be used, resulting in the group Lasso; or for the case of images, a penalty based on image gradient, giving the total variation penalty.

L_p norms for the $0 < p < 1$ case can also be used and give sparser solutions. Equation (3.2) is not a true norm in this regime and the resulting quasi-norm is non-convex. Notwithstanding these concerns, quasi-norms have been shown to be useful in a number of settings (Chartrand, 2007; Kaban and Durrant, 2008).

The relevance vector machine (RVM) uses the hierarchical construction of the Student's-t distribution as a prior for sparse learning by maximising the marginal likelihood, often referred to as type II maximum likelihood (Tipping, 2001). There also exists a number of greedy approaches to sparse learning which do not consider the L_1 norm, such as Iterative Hard Thresholding (Blumensath and Davies, 2008) or Rodeo (Lafferty and Wasserman, 2008), amongst others.

The construction of Bayesian methods congruent to a compound norm such as equation (3.30) is not straightforward. To do this, a prior would be specified using the Gibbs measure with the relevant energy function $E(x)$ being the compound norm. The Gibbs measure is :

$$p(x) = \frac{1}{Z}\exp\left(-E(x)\right). \tag{3.31}$$

This idea has already been used in the specification of the Laplace distribution, where the energy function $E(x) = \gamma\|x\|_1$, and the partition function $Z(\gamma)$ ensures normalisation. For the case of the fused Lasso, the energy function would be given by equation (3.30), and results in the normalising constant $Z(\alpha, \beta)$, where α, β are the regularisation parameters introduced in equation (3.30), but is intractable. If α and β are fixed then $Z(\alpha, \beta)$ is not needed for Bayesian inference. If we wish to learn α and

β from data then it is essential that we know what this normalisation constant is. Variational methods based on determining suitable bounds may be used in this case, such as the approach taken by Marlin et al. (2009) for a group sparse priors. There is a wide body of research for finding bounds for partition functions in the machine learning literature, and whose exploration will provide Bayesian interpretations of this interesting class of penalties.

Sparse optimisation methods are based mainly on the use of convex functions, hence the popularity of the L_1 norm. Submodular functions are the equivalent of convex functions in discrete optimisation. Since problems in sparsity begin as a discrete combinatorial problem, there is great interest in the use of submodular optimisation for sparse learning. The work on structured sparsity and the connections between submodular optimisation and sparsity by Bach (2010) represents some of the latest advances in this line of sparse learning.

3.6.2 Learning Compressed Sensing

Compressed sensing (or compressive sampling) (Candes et al., 2006; Donoho, 2006) is an immensely popular area of research. Compressed sensing methods allow reconstructions of data to be made from undersampled data and have the promise of providing significant reductions in measurement times at lower costs for a wide range of applications, from medical imaging to military surveillance. Compressed sensing, at its core, uses sparsity to make effective use of limited measurements by moving the measurement workload from the sensing apparatus to computation at reconstruction time.

Compressed sensing is understood by considering a two phase system consisting of an encoder and a decoder.

Encoder. The encoder describes the generative process by which samples are acquired, and consists of two components: a sparsity and a measurement component. These are expressed mathematically as:

$$\begin{aligned} &\textbf{Sparsity component:} && \mathbf{f} = \mathbf{\Psi}^*\mathbf{x} \\ &\textbf{Measurement component:} && \mathbf{y} = \mathbf{\Phi}\mathbf{f} \\ & && \therefore \mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}^*\mathbf{x}. \end{aligned} \tag{3.32}$$

The signal of interest $\mathbf{f} \in \mathbb{R}^N$ is said to be sparse in the basis given by $\mathbf{\Psi}$, where $\mathbf{x} = \mathbf{\Psi}\mathbf{f}$. $\mathbf{x}$ is the sparse representation of the signal. The sparsity basis is assumed to be known in most cases and could be common bases such as the Fourier or wavelet bases. Separate from this, is the measurement component with low dimensional samples $\mathbf{y} \in \mathbb{R}^M$ for $M < N$, and $\mathbf{\Phi}$ is the measurement

basis, which for compressed sensing is chosen as a random matrix.

Decoder. To recover the signal $\mathbf{f}$, the decoder solves an L_1 optimisation problem to determine the sparse vector $\mathbf{x}$, which is used with the known sparsity basis to recover the signal. Let $\mathbf{\Phi}' = \mathbf{\Phi}\mathbf{\Psi}^*$, then the optimisation is:

$$\min \|\mathbf{x}\|_1, \qquad \text{subject to } \mathbf{y} = \mathbf{\Phi}'\mathbf{x}. \tag{3.33}$$

The setup of the compressed sensing problem is very different to the general setup that we have discussed throughout this chapter. The models and results we have developed consider a high dimensional data set (D dimensional), and use the latent variable modelling approach to obtain a low dimensional representation (K dimensional with $K < D$). For compressed sensing, the opposite is true, a low dimensional data set (set of M undersampled measurements) is used to recover a high dimensional signal of interest (N dimensional, $N > M$). But, the components of both settings that focus on determining the sparse representations are identical. As such, the models discussed are immediately applicable to the compressed sensing problem of determining a sparse representation of observed data. In the sparse literature, the setup considered by compressed sensing is closely related to problem of learning an overcomplete representation (Lewicki and Sejnowski, 1998).

Bayesian approaches to compressed sensing have been considered by Ji et al. (2008) and Seeger (2008). Bayesian methods allow for noise in the measurements, which is not a setting considered in the theory of compressed sensing. In addition, information regarding the uncertainty of the reconstruction is obtained and Bayesian methods provide a means with which to decide when a sufficient number of measurements have been obtained (Ji et al., 2008). There is thus scope for much wider applications of the approaches to sparse learning presented in this chapter.

One additional advantage of the methods discussed in this chapter is the ability to learn the measurement matrix $\mathbf{\Phi}$ from the data as opposed to the use of a random matrix supported by the compressed sensing literature. This is an area of contention, since the randomly selected basis allows a non-adaptive, and hence fast approach to reconstruction that lies at the core of compressed sensing. But $\mathbf{\Phi}$ can be learnt in advance from from a large database of signals from the application domain, thus mitigating concerns regarding speed. In addition, Weiss et al. (2007) demonstrate that in the expected setting of signals with measurement noise, learning the basis $\mathbf{\Phi}$ can provide significant improvements in signal reconstruction. The results of Weiss et al. (2007) provide an initial motivation for a more concerted investigation of the applicability of sparse learning methods to compressed sensing and the development of highly scalable fast algorithms for the task.

3.6.3 Infinite Dimensional Settings

The two classes of sparse Bayesian priors considered: the continuous sparsity-favouring prior and the discrete mixture prior, were constructed using a finite K-dimensional latent variable. We can also consider infinite dimensional generalisations of these two classes of priors wherein the theory of Bayesian non-parametric methods applies. Bayesian non-parametric models are models on infinite dimensional parameter spaces, but which use only a finite subset of the parameter dimensions to explain a finite set of data (Orbanz and Teh, 2010). This is highly desirable, since it allows the model complexity to adapt to the data. One model specification problem that we have encountered thus far, is the specification of the number of latent factors. Using non-parametric methods, the number of latent factors can be inferred from the data, rather than needing to be specified beforehand.

The continuous sparsity-favouring priors discussed were based on the formulation of a K-dimensional Gaussian scale-mixture representation, where the choice of the mixing density gave rise to several priors with properties amenable for sparse Bayesian learning. The Normal-Gamma and the Normal Inverse-Gaussian were two such examples. If the properties of these distributions are considered as $K \to \infty$, then it can be shown that the resulting priors are Lévy processes. For the Normal-Gamma, the infinite dimensional analogue was shown by Caron and Doucet (2008) to be a variance-gamma process, where the parameters are the jumps of this Lévy processes. Similarly, the Normal-Inverse Gaussian can be shown to correspond to an infinite variation process. More recently, Polson and Scott (2010) showed that in general, scale-mixture distributions with mixing densities that are self-similar (closed under addition) are Lévy processes.

The spike-and-slab prior was constructed by considering a K-dimensional Bernoulli vector with Beta priors. Taking the limit as $K \to \infty$ gives rise to a non-parametric prior known as the Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2006). The design of sparse models based on the IBP, corresponding to a non-parametric version of the spike-and-slab model, was presented by Knowles and Ghahramani (2007, 2010). The IBP also has the practical advantage of allowing the latent dimensionality K to be learnt directly from the data.

The philosophical dichotomy between strong and weak sparsity implied by the two classes of prior, remains though. These two classes of non-parametric sparse priors have not been discussed together in any detail in existing work, leaving scope for such a treatment. An interesting line of thought is the unification of the classes of discrete mixture, and continuous priors based on scale mixtures, with this idea having been recently elaborated upon by Polson and Scott (2010).

3.6.4 Sparsity: Assumption or Hypothesis

Throughout this chapter we have considered sparsity as an *assumption*, a property inherent to the data. This assumption is not unreasonable, particularly given the nature of the systems under study, as in the motivating examples described in section 3.1. An ideal setting would avoid this assumption and view sparsity as a *hypothesis* to be tested. This view would require mechanisms by which to quantify the information content of observed data and allow the practitioner to conclude whether a given data set warrants the use of sparse methods or not.

The spike-and-slab prior is also used in the area of Bayesian *multiple testing*, though not with the motivation of learning sparse representations. Bayesian multiple testing employs sparsity as a means of simultaneously testing the hypothesis $H_{0d} : \omega_d = 0$ against $H_{1d} : \omega_d \neq 0$ for $d = 1, \dots D$ for a parameter vector of interest $\boldsymbol{\omega}$. Such a Bayesian 'testimation' procedure (Abramovich et al., 2007) can be achieved in many ways, but the spike-and-slab has proven popular for this task (Scott and Berger, 2006). The use of Bayesian methods in the multiple testing scenario is promoted particularity because it allows for the data-driven characterisation of underlying sparsity levels. Thus, the idea of sparsity as a hypothesis has been considered, though the literature in the two areas do not often coincide.

The Bayesian non-parametric approach based on the IBP discussed above also provides a means of achieving this simultaneous estimation and sparsity characterisation, particularly for the case of latent variable models discussed in this thesis. The spike-and-slab nature of the IBP, and the inherent ability to adjust the latent dimensionality to that supported by the data make these methods especially appealing. As previously referenced, some work already exists (Knowles and Ghahramani, 2007), but there seems scope for both a wider study of non-parametric Bayesian multiple testing (Ghosal and Roy, 2009) as well as stronger links between multiple testing and sparse learning.

3.6.5 Re-thinking the Slab Distribution

The spike-and-slab is constructed in most work using a Gaussian distribution for the slab, as is the case in this chapter. This is a suitable default choice, but there remains little guidance as to choosing this slab distribution. As discussed for the continuous sparsity priors, the tail behaviour of these priors is of central importance. It may be that more robust inferences can be made in the spike-and-slab setting with a heavy-tailed slab rather than a Gaussian. Johnstone and Silverman (2004) provide the first analysis in this regard by considering a Laplace slab, as well as a slab based on a scale-mixture prior. The use of a heavy tailed slab is also alluded to in other

works (Griffin and Brown, 2010). Knowles and Ghahramani (2007) consider the non-parametric Bayesian setting using the Indian Buffet Process with a Laplace slab. The results of this work do not suggest that much is gained by using the Laplace slab, but this may be attributable to deficiency of the slab choice. Thus, a much more systematic study of alternative slab distributions is required.

3.7 Sparse Learning in Context

The development of modern views of sparsity are grounded in a number scientific communities. The earliest thoughts regarding L_1 penalisation are traced to geo-physics with the work of Claerbout and Muir (1973) in conjunction with absolute error loss functions, and Santosa and Symes (1986) using the least-squares loss function. This work was followed by early results for the L_1 minimisation problem in statistics by Donoho and Stark (1989), leading up to the introduction of the LASSO by Tibshirani (1996) for penalised regression. The development of the LASSO saw the establishment of the L_1 norm as a means of introducing sparsity in many regression problems, and soon saw the widespread application of these ideas in more specialised models such as the fused Lasso (Tibshirani et al., 2005), group Lasso (Yuan and Lin, 2006), L_1 regularised logistic regression (Lee et al., 2006b), and in wider generalised linear models (Park and Hastie, 2007) as well as a strong focus on the development of more efficient algorithms for learning (Efron et al., 2004; Lee et al., 2006b; Schmidt et al., 2007; Duchi and Singer, 2009).

Concurrently in Bayesian learning, the ideas for sparsity were developed for variable selection with the introduction of the spike-and-slab prior by Mitchell and Beauchamp (1988) and subsequent authors (George and McCulloch, 1993; Ishwaran and Rao, 2005). Bayesian methods for sparse regression have since been considered at length, with O'Hara and Sillanpäa (2009) provide a review of Bayesian methods for variable selection.

Sparse methods soon found application in a number of scientific areas including source separation, image coding and in the new field of compressed sensing (Candes et al., 2006; Donoho, 2006). We highlighted the connections between compressed sensing and sparsity in the latent variable modelling framework in section 3.6.2. Sparsity is invaluable in learning the connectivity structure in graphs (Meinshausen and Bühlmann, 2006; Lee et al., 2006a), in high dimensional data analysis in genomics (Srebro and Jaakkola, 2001; Carvalho et al., 2008) and in financial modelling (Brodie et al., 2009; Carvalho et al., 2010b).

In unsupervised learning, the focus of this chapter, the optimisation of the L_1 norm has lead to various versions of sparse PCA (Zou et al., 2004; d'Aspremont

et al., 2005; Zass and Shashua, 2006). The wide body of literature on matrix factorisation is also indirectly related (Airoldi et al., 2008; Srebro et al., 2005a). These methods may yield fairly sparse factors, but as a by-product rather than by construction. Methods for matrix factorisation designed with sparsity in mind have also been considered such as those by Srebro and Jaakkola (2001); Dueck and Frey (2004). Independent Components Analysis (ICA) (Jutten and Hérault, 1991; Common, 1994) is very relevant and is a broad term used to refer to models similar to factor analysis, but where the latent distribution is non-Gaussian. ICA now has a wide associated literature and closely related to the problem of blind source separation where sparsity is useful in separating speech signals from a mixed signal sources. For ICA, the Laplace distribution or other heavy tailed distributions for the latent variables are commonly used. This chapter has contributed to this broad unsupervised model exploration by developing both the optimisation and Bayesian approaches for sparsity in the generalised latent variable model setting, exploring the various classes of priors available, developing new inference strategies, and comparing and contrasting these methods for the first time. At the same time that we developed the model for sparse EPCA in this chapter, Lee et al. (2009) described a very similar idea, but in the context of a model for semi-supervised learning.

Continuous scale mixture priors are in widespread use: the Laplace is well studied (Seeger et al., 2007; Park and Casella, 2008); the Normal-Jeffrey's prior is discussed by Figueiredo and Member (2003), the Normal-Gamma and the Normal-Inverse Gaussian are described by Caron and Doucet (2008) and the Normal-Exponential Gamma is described by Griffin and Brown (2005). Discrete mixtures are discussed for genomic applications by West (2003) and Carvalho et al. (2008), considering the use of spike-and-slab priors to introduce sparsity in Bayesian factor regression models. This model combines latent factors with a set of response variables and sparsity included in the factor loadings (parameters $\mathbf{\Theta}$, rather than $\mathbf{V}$ as used in this chapter) for the problem of gene expression genomics. Inference in these factor regression models is achieved through a similar paired sampling of latent indicator and continuous variables as used in section 3.3.3.2. The work of Polson and Scott (2010) represents the latest thinking in sparse learning, relating scale-mixture priors and the specification of penalty functions to Lévy processes.

3.8 Summary

In this chapter we introduced new models that include sparsity in generalised latent variable models, providing an important new class of sparse models for data best modelled by distributions other than the Gaussian. We provided new sampling methods for sparse Bayesian learning using both continuous sparsity-favouring priors and the spike-and-slab distribution. At the same time, this chapter has provided the first comparison of optimisation and Bayesian approaches to sparsity. The methods were compared on both synthetic and real world data comparing the same models with sparsity in the latent factors $\mathbf{V}$, and examining their predictive performance on held out data. The spike-and-slab model was shown to provide the best predictive performance on all data sets, a success attributed to its ability to learn the underlying sparsity supported by the data, while not enforcing shrinkage on parameters of interest.

We have also attempted to expose some of the more subtle issues relating to sparse learning, such as considering strong and weak sparsity, optimisation and Bayesian learning, thinking about penalties beyond the L_1 norm and examining the connections to various other fields. The discussion also provided a number of areas for future work that will enhance the understanding and practical future application of sparse methods.

Chapter 4

Binary PCA by Latent Gaussian Dichotomisation

This chapter develops a simple and novel approach for modelling *correlated binary variables*. We review existing approaches for learning correlation in models, and describe a method for constructing correlated binary variables known as Gaussian dichotomisation. The basic idea is to dichotomise (threshold) a correlated Gaussian latent variable, resulting in a correlated binary vector. We derive moment-matching equations and develop an efficient algorithm to learn the distribution of the latent Gaussian, using this algorithm as part of a new method for binary PCA.

4.1 Generating Correlated Binary Variables

Data sets from a vast array of application areas including social networks, the web, information retrieval, topic modelling and collaborative filtering, appear as large, sparse binary data. There is a great interest in being able to learn and use correlation when making predictions and recommendations based on these binary data sets. The collaborative filtering task of providing movie recommendations, such as the popular Netflix challenge (Netflix, 2009), is based on a binary data set of users' viewing history. Knowledge of the correlation between movies aids in the suggestion of new movies and can be particularly useful for users lacking an established viewing history. In topic modelling applications, it is reasonable to expect that articles on genetics are correlated with articles on disease and health, but unlikely with x-ray astronomy (Blei and Lafferty, 2005). Models with correlation allow the natural relationships expected in real data to be accounted for and are advantageous since they allow fine-grained structure in data to be learnt, as well as more robust inferences to be made by sharing statistical power between measurements.

There are a number of approaches that can be used to generate correlated binary variables. Correlation can be introduced by incorporating an additional set of hierarchical latent variables. In this setting, observed binary variables $\mathbf{x}$ are assumed to be independent given additional latent variables $\mathbf{v}$ i.e. $x_i \perp\!\!\!\perp x_j|\mathbf{v}$, and marginalisation of the latent variables induces correlation in the binary variables $\mathbf{x}$. This approach provides a means of constructing wide classes of algorithms for learning based on maximum-likelihood or fully Bayesian inference, and is used in a number of settings. Factor analysis, which has been widely discussed in this thesis, is one such example of inducing dependencies in observed data using a set of hierarchical latent variables. The factor analysis model for observed data $\mathbf{x}$ with latent variables $\mathbf{v}$ and the $D \times K$ factor loadings matrix $\mathbf{\Theta}$ is:

$$\mathbf{x} = \mathbf{\Theta}\mathbf{v} + \boldsymbol{\epsilon} \tag{4.1}$$

$$\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi}). \tag{4.2}$$

$\mathbf{\Psi}$ is forced to be diagonal, either $\mathbf{\Psi} = \sigma^2\mathbf{I}$ as used in equation (2.2), or $\mathbf{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_D)$. Marginalisation of the latent variables gives the covariance of the observed data as $\mathbf{\Sigma} = \mathbf{\Theta}\mathbf{\Theta}^\top + \mathbf{\Psi}$. Factor analysis encodes correlation between the elements of a high dimensional vector $\mathbf{x}$, by dependence on a set of lower dimensional latent variables $\mathbf{v}$. These ideas can be extended to modelling correlation between binary variables and latent variables, as has been demonstrated by Doshi-Velez and Ghahramani (2009); Li and McCallum (2006) for correlated non-parametric latent feature models and in models based on sigmoid networks and deep learning (Hinton et al., 2006).

Correlation can be encoded directly into models, using distributions parameterised in terms of means and covariances. The correlated topic model (Blei and Lafferty, 2005) makes use of a logistic-Normal distribution and transforms draws from a normal distribution using the logistic sigmoid function, to represent correlation amongst proportions of topics in the model, where the correlation is encoded through the covariance matrix of the Gaussian distribution. For binary data, this direct encoding approach has been popular with several methods available that allow correlated binary data to be generated based on the specification of the first two moments (Emrich and Peidmonte, 1991; Qaqish, 2003). We explore this direct approach further in this chapter, using an approach known as Gaussian dichotomisation.

4.2 Gaussian Dichotomisation

Gaussian dichotomisation, first discussed by Pearson (1909), is the process of generating a dichotomous or binary variable $x \in \{0, 1\}$, by thresholding a Gaussian latent variable v at zero. By combining an underlying regression model with covariates $\mathbf{y} \in \mathbb{R}^p$ for the latent response v, this process can be described as:

$$v = \mathbf{y}^\top \boldsymbol{\beta} + \lambda \tag{4.3}$$

$$x = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

where λ is the noise level. The model corresponds to a probit model if λ has a Gaussian distribution, a logit model if the noise is a logistic distribution and the complementary log-log model if the noise is a Gumbel distribution.

Thus a univariate dichotomisation simply involves the generation of an underlying univariate Gaussian distribution and thresholding realisations from this Gaussian at zero to obtain binary variables. The multivariate case is obtained by thresholding realisations from a multivariate Gaussian distribution instead. Consider generating a correlated binary vector $\mathbf{x} \in \{0, 1\}^D$ with given means r_i and pairwise covariance Σ_{ij} for $i, j = 1, \ldots, D$:

$$\mathbf{x} \sim \mathcal{CB}(\mathbf{x}|\mathbf{r}, \boldsymbol{\Sigma}) \tag{4.5}$$

$$r_i = \mathbb{E}[x_i] \tag{4.6}$$

$$\Sigma_{ij} = \mathbb{E}[x_i x_j] - \mathbb{E}[x_i]\,\mathbb{E}[x_j]\,. \tag{4.7}$$

Assume that the correlated vector $\mathbf{x}$ has been generated by dichotomisation, i.e. that there exists a latent Gaussian vector, which after thresholding generates the observed data $\mathbf{x}$.

$$\mathbf{v} \sim \mathcal{N}(\mathbf{v}|\boldsymbol{\gamma}, \boldsymbol{\Lambda}), \tag{4.8}$$

$$x_i = \mathbb{I}(v_i > 0) \qquad (i = 1, \ldots, D). \tag{4.9}$$

The dichotomisation of the latent Gaussian $\mathcal{N}(\mathbf{v}|\boldsymbol{\gamma}, \boldsymbol{\Lambda})$ changes the moments of the resulting distribution, but these changes can be determined and accounted for. By matching the moments of the latent Gaussian distribution (4.8) and the desired correlated binary distribution (4.5), the mapping between the two distributions can be established as:

$$r_i = \Phi(\gamma_i) \tag{4.10}$$

$$\Sigma_{ii} = \Phi(\gamma_i)\Phi(-\gamma_i) \tag{4.11}$$

$$\Sigma_{ij} = \Psi(\gamma_i, \gamma_j, \Lambda_{ij}) = \Phi_2(-\gamma_i, -\gamma_j, \Lambda_{ij}) - \Phi(\gamma_i)\Phi(\gamma_j), \tag{4.12}$$

assuming that $\Lambda_{ii} = 1$ without loss of generality, Φ is the standard cumulative univariate Gaussian and $\Phi_2(x, y, \rho)$ is the bivariate cumulative Gaussian with correlation ρ. These equations are noted by a number of authors including Leisch et al. (1998); Cox and Wermuth (2002) and Macke et al. (2009). For the case of a bivariate Gaussian, the resulting assignment of mass to the four binary outcomes after dichotomisation is illustrated in figure 4.1a.

4.2.1 Deriving the Moment Matching Equations

The derivation of equations (4.10) – (4.12) is not given in existing work, so it is instructive to consider their derivation here. The means r_i are given by:

$$\begin{aligned} r_i &= p(x_i = 1) = p(v_i > 0) = p(v_i - \gamma_i > -\gamma_i) = p(v_i - \gamma_i < \gamma_i) \\ &= \Phi(\gamma_i), \end{aligned} \tag{4.13}$$

which proves equation (4.10), and the second-last step follows from the symmetry of the Gaussian distribution. The variance Σ_{ii} of the binary variable is:

$$\begin{aligned} \Sigma_{ii} &= r_i(1 - r_i) = \Phi(\gamma_i)(1 - \Phi(\gamma_i)) \\ &= \Phi(\gamma_i)\Phi(-\gamma_i), \end{aligned} \tag{4.14}$$

where the last step is obtained by recalling that $1 - \Phi(\gamma_i) = \Phi(-\gamma_i)$ for the standard Gaussian, and proves equation (4.11). To compute the covariances Σ_{ij}, recall that the correlation between i and j is defined as $\rho = \frac{\Lambda_{ij}}{\Lambda_{ii}\Lambda_{jj}} = \Lambda_{ij}$ since Λ_{ii} is assumed to be unity, and that this correlation is bound between [-1,1].

$$\begin{aligned} \Sigma_{ij} &= cov(x_i, x_j) = p(x_i = 1, x_j = 1) - p(x_i = 1)p(x_j = 1) \quad \forall i \neq j \\ &= p(x_i = 1, x_j = 1) - r_i r_j. \end{aligned}$$

The second term in the above equation can be computed using the previous result of equation (4.13). The first term is:

$$\begin{aligned} p(x_i = 1, x_j = 1) &= p(v_i > 0, v_j > 0) = p(v_i - \gamma_i > -\gamma_i, v_j - \gamma_j > -\gamma_j) \\ &= p(v_i - \gamma_i > -\gamma_i, v_j - \gamma_j > -\gamma_j) \\ &= \int_{-\gamma_i}^{\infty} \int_{-\gamma_j}^{\infty} \mathcal{N}(v_i - \gamma_i, v_j - \gamma_j, \Lambda_{ij}) \\ &= \Phi_2(-\gamma_i, -\gamma_j, \Lambda_{ij}), \end{aligned}$$

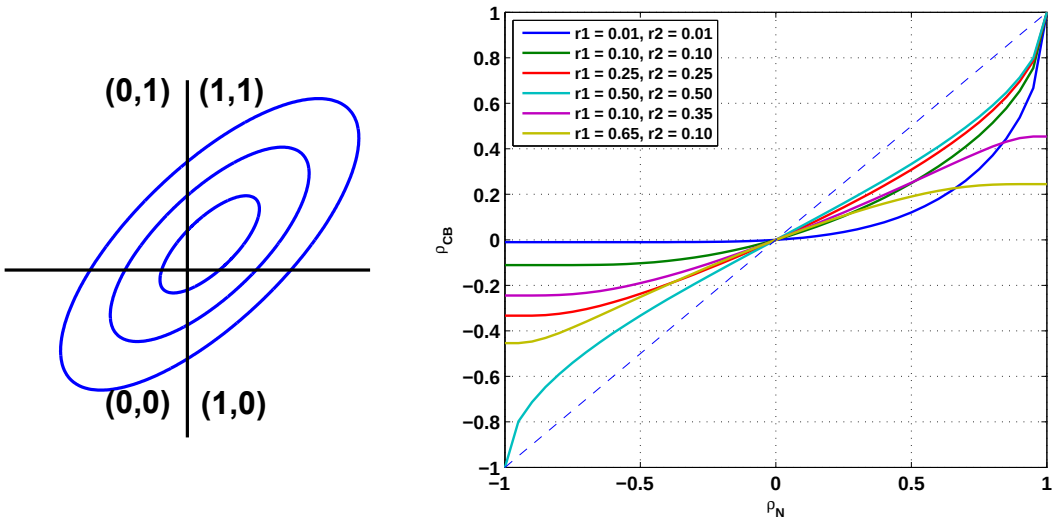


Figure 4.1: (a) Assignment of binary variables by dichotomisation of the bivariate Gaussian distribution. (b) Relationship between the correlation coefficient for the Binary random variables and the latent Gaussian response, ρ_{CB} and ρ_N respectively.

where $\Phi_2(\cdot)$ is the bivariate Gaussian distribution with correlation Λ_{ij} in the integration step. Combining terms equation (4.12) can be verified.

$$\begin{aligned} \therefore \Sigma_{ij} &= \Phi_2(-\gamma_i, -\gamma_j, \Lambda_{ij}) - \Phi(\gamma_i)\Phi(\gamma_j) \\ &= \Psi(\gamma_i, \gamma_j, \Lambda_{ij}). \end{aligned} \tag{4.15}$$

4.2.2 Solving the Equations

Solving the equations for the γ_i and Λ_{ij} can be achieved by inverting the equations (4.10) – (4.12). Given a desired mean vector $\mathbf{r}$, we obtain the underlying Gaussian mean using:

$$\gamma_i = \Phi^{-1}(r_i). \tag{4.16}$$

This is the probit function and can be expressed in terms of error functions for which the inverse can be computed (based on known rational approximations). Determining Λ_{ij} requires solving :

$$\Sigma_{ij} - \Psi(\gamma_i, \gamma_j, \Lambda_{ij}) = 0,$$

which can be solved by bisection since the result in bound to the region [-1,1] and the function is monotonic in Λ_{ij}. The monotonicity can be seen since $\Phi_2(x, y, \rho)$ is strictly increasing in ρ for a given x, y. Once γ and $\mathbf{\Lambda}$ have been determined, then sampling a correlated binary vector is as straightforward as sampling from a multivariate Gaussian distribution.

Figure 4.1b shows the relationship between the correlation for the binary random variables ρ_{CB}, and the correlations for the latent Gaussian ρ_N, for various mean probabilities. From the figure, it can be seen that dichotomisation is a process that diminishes correlation. For low marginal probabilities r_i, this effect is most noticeable, where for a wide range of correlations for the latent Gaussian variable, the resulting binary correlation is constant (dark blue curve). The effect is least noticeable in the symmetric case with $r_1 = r_2 = 0.5$ (turquoise curve).

4.2.3 Restrictions on the Covariance Matrix

Caution must be taken when using Gaussian dichotomisation, since every symmetric positive definite matrix that can be specified can not be used as a covariance matrix for a correlated binary distribution. Restrictions on the covariance matrix are required to ensure that none of the implied pairwise probabilities are negative. For two binary random variables X and Y with means p and q respectively, the covariance between the two binary variables is bound by:

$$\max\{-pq, -(1-p)(1-q)\} \quad \leq cov(X,Y) \leq \quad \min\{(1-q)p, (1-p)q\}, \quad (4.17)$$

where these bounds can be shown by looking at the bounds on the joint and marginal probabilities in the definition of the covariance between X and Y.

For this case of two random variables, the bounds are well known (Leisch et al., 1998; Macke et al., 2009). For more general settings with 3 or more random variables, conditions for validity are shown by Chaganty and Joe (2006), who show that the multivariate probit, which is the construction used in Gaussian dichotomisation, has a wider coverage of covariance matrices compared to a number of other methods for generating correlated binary variables. Gaussian dichotomisation can be used to check the validity of a specified binary covariance matrix $\mathbf{\Sigma}$: the covariance matrix of an underlying Gaussian distribution $\mathbf{\Lambda}$ is computed using the dichotomisation equations. If this covariance is positive-definite, then the initial covariance matrix is a valid covariance matrix for binary variables.

4.2.4 Evaluating the Probability of a Binary Vector

Evaluating the probability of a correlated binary vector obtained by Gaussian dichotomisation requires the evaluation of the following integral:

$$p_{CB}(\mathbf{x}) \quad = \quad \frac{1}{(2\pi)^{N/2}|\mathbf{\Lambda}|^{1/2}} \int_{a_1}^{b_1} \cdots \int_{a_p}^{b_p} \exp\{-(\mathbf{x}-\boldsymbol{\gamma})^\top \mathbf{\Lambda}^{-1}(\mathbf{x}-\boldsymbol{\gamma})\}, \quad (4.18)$$

where the limits of integration are:

$$a_i = 0, \quad b_i = \infty \quad \text{if} \quad x_i = 0;$$
$$a_i = -\infty, \quad b_i = 0 \quad \text{if} \quad x_i = 0.$$

This integral must be evaluated numerically and for which a number of solutions exist. The method of Genz (1992) (QSIMMVNDV) is one way, which may be more efficient than others. The theses of Minka (2001) and Cunningham (2009) also provide new tools for evaluating these Gaussian probabilities based on the method of expectation propagation (EP).

4.2.5 Sampling from a 3-dimensional Correlated Binary Vector

As a simple example, consider generating samples from a 3-dimensional correlated binary vector $\mathbf{x} = [x_1, x_2, x_3]$ with mean $\mathbf{r}$ and covariance $\mathbf{\Sigma}$, with the following marginal and pairwise probabilities:

$$p(x_1 = 1) = 0.25; \quad p(x_2 = 1) = 0.5; \quad p(x_3 = 1) = 0.75;$$

$$p(x_1 = 1, x_2 = 1) = 0.1; \quad p(x_1 = 1, x_3 = 1) = 0.125; \quad p(x_2 = 1, x_3 = 1) = 0.4.$$

The pairwise probabilities imply the following covariance matrix or equivalent correlation matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} 0.1875 & -0.025 & -0.0625 \\ -0.025 & 0.2500 & 0.025 \\ -0.0625 & 0.025 & 0.1875 \end{bmatrix} \qquad \boldsymbol{\rho}_{CB} = \begin{bmatrix} 1 & -0.1155 & -0.3333 \\ -0.1155 & 1 & 0.1155 \\ -0.3333 & 0.1155 & 1 \end{bmatrix}.$$

Using the moment matching equations (4.10) – (4.12) the mean $\boldsymbol{\gamma}$ and the correlation $\mathbf{\Lambda}$ of the latent Gaussian is:

$$\boldsymbol{\gamma} = [-0.6745, 0.00, 0.6745]$$

$$\mathbf{\Lambda} = \begin{bmatrix} 1 & -0.1965 & -0.5343 \\ -0.1965 & 1 & 0.1965 \\ -0.5343 & 0.1965 & 1 \end{bmatrix}.$$

Figure 4.2 shows 50 samples of the correlated binary vector generated by Gaussian dichotomisation. The empirical correlation and probabilities obtained using 10,000 samples are given below, and are very close to the true values, verifying the correctness of the sampling.

$$\bar{p}(x_1 = 1) = 0.2441; \quad \bar{p}(x_2 = 1) = 0.5004; \quad \bar{p}(x_3 = 1) = 0.7521;$$

$$\bar{p}(x_1 = 1, x_2 = 1) = 0.0976; \quad \bar{p}(x_1 = 1, x_3 = 1) = 0.1229; \quad \bar{p}(x_2 = 1, x_3 = 1) = 0.3997.$$

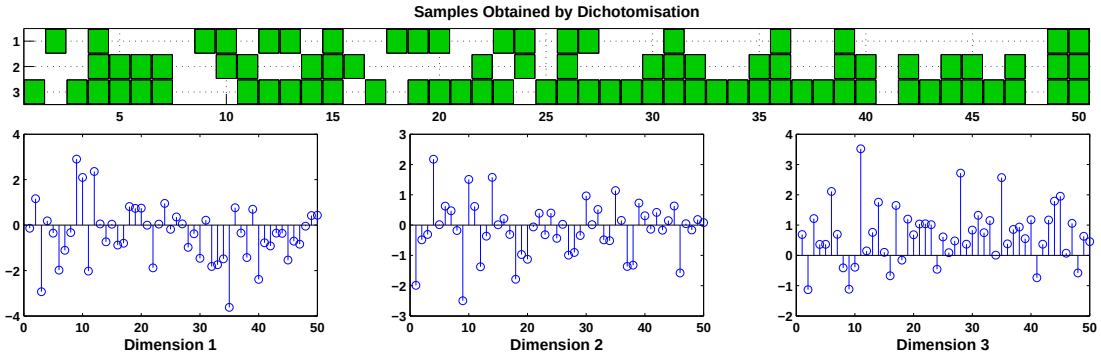


Figure 4.2: 50 correlated binary vectors obtained by Gaussian dichotomisation for the 3-dimensional example.

$$\bar{\mathbf{\Sigma}} = \begin{bmatrix} 0.1845 & -0.0246 & -0.0607 \\ -0.0246 & 0.2500 & 0.0234 \\ -0.0607 & 0.0234 & 0.1865 \end{bmatrix}.$$

4.3 The Principal Components Analysis of Binary Data

The process of Gaussian dichotomisation suggests a simple method for applying Principal Components Analysis (PCA) to binary data using the moment-matching equations (4.10) – (4.12). A simple algorithm is as follows:

1. Compute the empirical means $\mathbf{r}$ and covariance matrix $\mathbf{\Sigma}$ from the observed binary data.
2. Compute the latent Gaussian correlation matrix $\mathbf{\Lambda}$ using equation (4.12).
3. Compute the principal components of $\mathbf{\Lambda}$ using the usual PCA algorithm. This involves diagonalisation of $\mathbf{\Lambda}$ and using the eigenvectors corresponding to the K-largest eigenvalues as the principal components. Efficient methods exist for computing the top K eigenvectors.

The ease of this approach is demonstrated by the following two examples.

Binary Digits. We used the USPS digits data set to demonstrate the behaviour of our PCA algorithm. The data consists of 1000 16×16 images of handwritten digit '9'. We ran PCA to find 2 principal components. The underlying projection onto the principal component space for the 2-dimensions is shown in figure 4.3, and gives a representation of the writing styles of the digit. The reconstruction of the images at four points in the style-space is shown on the left of the image.

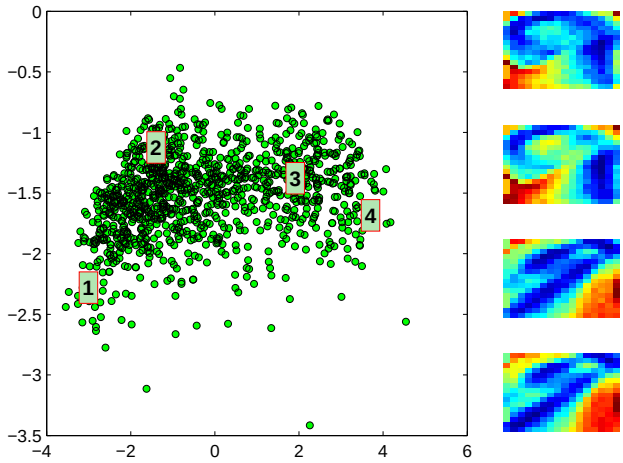


Figure 4.3: Visualisation of the embedding of the digit 9 data set. The images on the right show the image reconstructions at the numbered points in the latent space.

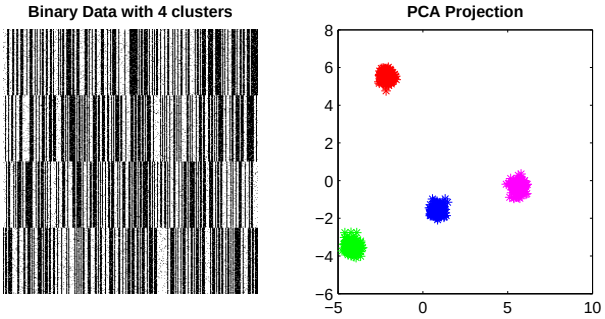


Figure 4.4: Visualisation of the embedding of data with four clusters. The images on the left show the original data and the right image is the projection of the 800 data points in the 2-dimensional space.

Clustered Synthetic Data. We generated synthetic binary data consisting of 4 clusters. Each observation is a 250 bit vector with 200 observations from each cluster. The underlying projection onto the principal component space for 2-dimensions is shown in figure 4.4 and shows clearly the existence of four clusters in the data.

4.4 Discussion

The probability of the multivariate binary vector obtained by Gaussian dichotomisation is given by:

$$p(\mathbf{x}_n|\boldsymbol{\gamma}, \boldsymbol{\Lambda}) = \int_{B_{nD}} \dots \int_{B_{n1}} \mathcal{N}(\mathbf{v}_n|\boldsymbol{\gamma}, \boldsymbol{\Lambda}) d\mathbf{v}_n, \tag{4.19}$$

where B_{nd} is in the interval $(0, \infty)$ if $x_{nd} = 1$ and the interval $(-\infty, 0)$ if $x_{nd} = 0$. Based on this construction, it can be seen that this model is a multivariate probit

construction. This latent variable construction can be used as the basis of Bayesian inference based on Gibbs sampling, which will require the simulation from univariate truncated Gaussian distributions and can be done efficiently. The details of this approach were first discussed by Albert and Chib (1993) in the univariate setting, followed by an analysis of the multivariate probit setting by Chib and Greenberg (1998). Full details of the sampling procedure are deferred to that work. In an alternative use, we have used the multivariate probit construction in this chapter with moment-matching to obtain the Gaussian dichotomisation.

The specification described here was based on the use of dependence information up to the second order, i.e. information of the covariance between binary elements. Higher-order correlation can be considered and a general specification for higher-order dependence between binary variables was given by Bahadur (1961). For a binary vector $\mathbf{x} = [x_1, \ldots, x_n], x_i \in \{0, 1\}$, define means α_i and the standardised scores z_i:

$$\alpha_i = \mathbb{E}(x_i) = p(x_i = 1) \tag{4.20}$$

$$z_i = \frac{x_i - \alpha_i}{\sqrt{\alpha_i(1-\alpha_i)}}. \tag{4.21}$$

The correlation between parameters of order 2 and higher are then defined as:

$$r_{ij} = \mathbb{E}(z_i z_j) \qquad i < j, \tag{4.22}$$

$$r_{ijk} = \mathbb{E}(z_i z_j z_k) \qquad i < j < k, \tag{4.23}$$

$$\ldots \tag{4.24}$$

$$r_{12\ldots n} = \mathbb{E}(z_1 z_2 \ldots z_n). \tag{4.25}$$

Based on these definitions, Bahadur's joint distribution for $\mathbf{x}$ is:

$$p(\mathbf{x}) = p_{[1]}(\mathbf{x}) \cdot f(\mathbf{x}) \tag{4.26}$$

$$p_{[1]}(x) = \prod_{i=1}^{n} \alpha_i^{x_i}(1-\alpha_i)^{1-x_i} \tag{4.27}$$

$$f(x) = 1 + \sum_{i<j} r_{ij} z_i z_j + \sum_{i<j<k} r_{ijk} z_i z_j z_k + \ldots + r_{12\ldots n} z_1 z_2 \ldots z_n, \tag{4.28}$$

where $p_{[1]}(\mathbf{x})$ is the joint probability assuming that all x_i are independent. The proof of this formulation is given by Bahadur (1961). While this is an attractive specification which takes into account higher order moments, it is computationally infeasible for correlation-orders greater that 2 or 3. This implies that one must assume that all higher order correlations are zero, thus this type of complete specification is also limited. Similar to our experience with Gaussian dichotomisation, when higher order correlations are ignored, the correlation parameters are not free to range between [-1, 1] to ensure the validity of the probability distribution that is defined. There is a

great deal of interest in learning higher order correlations and this remains an active area of research.

Gaussian dichotomisation can also be used to generate correlated Poisson vectors, using the property that Poisson vectors arise in limit of a Bernoulli process. This idea simply considers the generation of correlated binary vectors using Gaussian dichotomisation and summing the elements of the vectors to obtain a Poisson distributed vector with correlated components. This can be seen by considering the binary vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, each of length K, generated by Gaussian dichotomisation. Based on these vectors, counts are defined as:

$$y_i = \sum_{k=1}^{K} x_{ik}; \qquad y_j = \sum_{k=1}^{K} x_{jk}. \tag{4.29}$$

The set of y_i form a multivariate Binomial distribution. In the limit as $K \to \infty$, a multivariate Poisson distribution is obtained, which follows as an extension of the limiting property for the Binomial distribution. The covariance between any two entries is:

$$\mathrm{Cov}(y_i, y_j) \;=\; \mathrm{Cov}\left(\sum_{k=1}^{K} x_{ik}, \sum_{k=1}^{K} x_{jk}\right) = \sum_{k=1}^{K} \mathrm{Cov}\left(x_{ik}, x_{jk}\right). \tag{4.30}$$

Denoting $p(x_{ik} = 1) = m$ and $p(x_{jk} = 1) = n$, then a lower bound on the covariance can be given as:

$$\mathrm{Cov}(y_i, y_j) \;=\; \sum_{k=1}^{K} p(x_{ik} = 1, x_{jk} = 1) - mn \geq \sum_{k=1}^{K} -mn. \tag{4.31}$$

In the limit that $K \to \infty$, the individual $p(x_{ik}) \to 0$, which implies that the lower bound on the covariance approaches zero from below. Thus, negative correlation between elements of correlated Poisson vectors obtained by Gaussian dichotomisation, is not possible. More general settings for generating correlated count vectors can be obtained by extending the ideas of Chib and Greenberg (1998) for multinomial responses using the multivariate probit construction, with such an approach described by Macke et al. (2009).

4.5 Gaussian Dichotomisation in Context

The idea of Gaussian Dichotomisation stems from the ideas of Pearson (1909), and appears in the literature under many names such as dichotomisation, thresholding or clipping. The exploration of the moment-matching equations discussed here and the inherent limitations of the method is a more recent development. The earliest

description is the short paper by Emrich and Peidmonte (1991) and independently followed by the working paper of Leisch et al. (1998). The most influential of these descriptions though is given by Cox and Wermuth (2002). The derivation of the moment-matching equation is uncomplicated, but is not described in any of these existing works, and we have aimed to make the derivation explicit. Gaussian dichotomisation has been used by Bethge and Berens (2007) in the study of natural images, by a Macke et al. (2009) in the study of neural spike trains and in this chapter for binary PCA. We have used Gaussian dichotomisation to develop a new method for binary PCA. Restrictions on the validity of covariance matrices are thoroughly dealt with in the paper by Chaganty and Joe (2006).

Seemingly unrelated regression models (SUR) (Zellner, 1962) arise when we measure multiple responses for a group of items (or individuals). SUR models provide a method for including correlation between observations by considering a number of regression equations with correlated cross-equation error terms. Correlation between binary variables are described by in number of papers, such as those by Oman and Zucker (2001) and Qaqish (2003), as well as the approaches for introducing correlation based on hierarchical specifications (Blei and Lafferty, 2005; Hinton et al., 2006; Li and McCallum, 2006; Doshi-Velez and Ghahramani, 2009). Learning correlations in binary data is of great interest in many diverse research areas including computational neuroscience, social network analysis, collaborative filtering, computational advertising and data mining, with many advances in the construction of correlated binary distributions being made in these fields.

4.6 Summary

In this chapter we developed a moment-matching approach for learning correlation in binary data. Gaussian dichotomisation was described, which is based on thresholding an underlying Gaussian variable at zero to obtain a correlated binary vector. We derived fully the key equations for determining the moments of a latent Gaussian distribution and demonstrated the method using an example to highlight its features. Using Gaussian dichotomisation, we developed a simple algorithm for the principal components analysis of binary data. The multivariate probit construction that underlies Gaussian dichotomisation was also described in relation to popular Bayesian inference approaches in this setting.

The connections to generating multivariate counts were described briefly and is an interesting direction for further research in the development of models for count data. Another avenue for future work based on Gaussian dichotomisation is in the development of fast algorithms for the analysis of large and sparse binary data.

Chapter 5

Probabilistic Models for Tensor Factorisation

This chapter develops latent variable models for multi-way or tensor data. A latent variable model for tensor factorisation decomposes an observed tensor data set into latent factors that expresses the underlying information content in the data. When a data set has a natural multi-way structure, it is sensible to conserve this structure in the data analysis, as opposed to rearranging the data into a matrix or 2-way data set and employing matrix factorisation techniques. Using a tensor model, we are able to maintain spatial and other implicit structure in the natural representation of the data – structure that is often lost when representing a tensor as a matrix. In particular, we develop a probabilistic model for non-negative decompositions of tensor data, building on a cornerstone of tensor modelling, a model known as parallel factor analysis. We describe a hierarchical model for non-negative tensor factorisation and Bayesian approaches for inference using MCMC. We will also review the existing approaches for tensor modelling and lay a foundation for new developments in this important area of research.

5.1 From Matrix to Tensor Factorisation

Typical data analysis problems focus on a matrix $\mathbf{X}$ of the form: $observations \times measurements$, with each row $\mathbf{x}_n$ corresponding to independently collected data and the columns corresponding to the various measurements of relevance to the study. The models discussed in chapter 2 focused on a matrix factorisation, where intuitively, this is the process of decomposing the 2-way data set $\mathbf{X}$ into two factors $\mathbf{V}$ and $\mathbf{\Theta}$.

A tensor $\mathcal{X}$ is a multi-dimensional array often referred to as a P-way data set or a Pth-order tensor, with P array dimensions. A first-order tensor is a vector, a second-order tensor is a matrix and thereafter tensors are referred to as higher-order tensors. Each of the P array dimensions is called a mode of the tensor. There are numerous application areas that routinely produce data in tensor form. In time series modelling, the collection of data over time results in a natural 3-way data set of $observations \times measurements \times time$. In chemistry, fluorescence spectroscopy generates data of intensities arranged as *samples* $\times$ *emission wavelengths* $\times$ *excitation wavelengths* and is used to identify constituent compounds in testing samples. In neuroscience, neuro-imaging studies using MRI generate data of pixel values arranged as *subjects* $\times$ *sessions* $\times$ *voxels (horizontal co-ordinates* $\times$ *vertical co-ordinates* $\times$ *slice depth)*. Similarly to the matrix case, such P-way tensors can be decomposed into P factors for each of the tensor modes. In this chapter we explore the generalisation of latent variable models to multi-way or tensor data. Latent variable methods for tensors are of interest since this approach allows for concise descriptions of the data to be learnt, allows for prediction of any missing values and has the ability to take into account spatial and temporal relationships between observations. The construction of models for tensors follows as a natural extension of the modelling techniques employed in the preceding chapters of this thesis.

Tensors require an additional set of indices to distinguish each of the tensor modes. Throughout this chapter, a tensor will be denoted by a calligraphic symbol. A P-mode tensor $\mathcal{X}$ of dimensions $M_1 \times \ldots \times M_p \times \ldots \times M_P$, will be decomposed into P-factors with the pth factor denoted by the matrix $\mathbf{U}^{(p)}$, having dimensions $K \times M_p$. K is the number of the latent factors. The columns of $\mathbf{U}^{(p)}$ are $\mathbf{u}_r^{(p)}$ for $r = 1, \ldots, M_p$. The symbol $\otimes$ is the vector outer product. For $\mathcal{X} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$ with column vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ of size I, J, K respectively, $\mathcal{X}$ is a tensor of dimensions $I \times J \times K$ with elements $x_{ijk} = a_i b_j c_k$. The tensor obtained in this manner is referred to as a third-order rank-one tensor.

5.1.1 Models for Multi-way Data

Modelling approaches for tensors can be grouped into two broad classes. The *CP decompositions* are models based on the polyadic representation of a tensor, i.e. expressing the tensor as the sum of a finite number of rank-one tensors. This class of models is also referred to as canonical decomposition (CANDECOMP) or as parallel factor analysis (PARAFAC) (Harshman, 1970), hence the joint naming CP. The *Tucker decompositions* (Tucker, 1966) are a form of higher order principal components analysis, sometimes referred to as higher order SVD, N-mode factor analysis, or N-mode principal components analysis (Kolda and Bader, 2007; Acar and Yener, 2009).

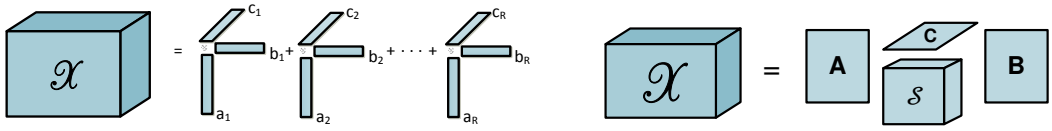


Figure 5.1: Approaches to tensor decomposition for 3-way arrays: (a) CP decomposition, (b) Tucker decomposition.

CP Decompositions. Consider a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. The tensor can be approximated using the following decomposition:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \tag{5.1}$$
$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \quad \text{(in elementwise form)}$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$, for $r = 1, \cdots, R$ for some positive integer R. This reconstruction is shown pictorially in figure 5.1a. It is sometimes useful to represent this using the shorthand $\mathcal{X} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where $\mathbf{A}$ is a *factor matrix* referring to the combination of the vectors from the rank-one components, i.e. $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_R]$, and likewise for $\mathbf{B}$ and $\mathbf{C}$. While the definition of equation (5.1) has been restricted to 3-way arrays for clarity, the definition can be easily extended to general P-mode tensors. In general, the factor matrices are not subject to any constraints, and variations of the CP model can be obtained by imposing additional constraints on the model factors and by considering different types of algorithms for learning.

Tucker Decompositions. For an $I \times J \times K$ tensor $\mathcal{X}$, the Tucker model is a decomposition of the form of equation (5.2) and is shown pictorially in figure 5.1b.

$$\mathcal{X} = \sum_{l=1}^{R_1} \sum_{m=1}^{R_2} \sum_{n=1}^{R_3} \sigma_{lmn} (\mathbf{a}_l \otimes \mathbf{b}_m \otimes \mathbf{c}_n) \tag{5.2}$$
$$x_{ijk} = \sum_{l=1}^{R_1} \sum_{m=1}^{R_2} \sum_{n=1}^{R_3} \sigma_{lmn} a_{il} b_{jm} c_{kn} \quad \text{(in elementwise form)}$$

where $i = 1 \ldots, I, j = 1 \ldots, J, k = 1 \ldots, K$. Here, $\mathbf{a}_l \in \mathbb{R}^I$, $\mathbf{b}_m \in \mathbb{R}^J$ and $\mathbf{c}_n \in \mathbb{R}^K$ for all l, m, n and $R_1 \leq I$, $R_2 \leq J$, $R_3 \leq K$, are the number of components (i.e. columns) in the factor matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ respectively. The tensor $\mathcal{S} = (\sigma_{lmn}) \in \mathbb{R}^{R_1 \times R_2 \times R_3}$, is called the *core tensor*, and its entries show the level of interaction between the different components. If R_1, R_2, R_3 are smaller than I, J, K, then the core tensor can be thought of as a compressed version of $\mathcal{X}$. The CP model is a special case of the Tucker decomposition in which $R_1 = R_2 = R_3 = R$ and the core tensor is equal to the identity tensor. In the general Tucker setting, there are no constraints on the vectors $\mathbf{a}_l, \mathbf{b}_m, \mathbf{c}_n$, however one may impose constraints if needed. If the $\mathbf{a}_l, \mathbf{b}_m, \mathbf{c}_n$ are columns from orthogonal matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, then the Tucker decomposition is known as the Higher-Order Singular Value Decomposition (HOSVD) (Lathauwer et al., 2000).

5.1.2 Learning with Non-negativity Constraints

Non-negativity constraints are one popular type of restriction on the tensor factors, with numerous applications of non-negative models in food science and image processing. Non-negativity is a natural constraint in many application areas: when data reflects colour intensities, counts or spectral amplitudes, negative numbers have no physical interpretation. Non-negativity constraints also have the tendency to generate sparse representations as a by-product (Lee and Seung, 1999), which enhances the interpretability of these models. The CP class of models is used for a P-way tensor of dimensions $M_1 \times \ldots \times M_P$:

$$\begin{aligned} &\mathcal{X} \approx \textstyle\sum_{k=1}^{K} \mathbf{u}_k^{(1)} \otimes \mathbf{u}_k^{(2)} \otimes \cdots \otimes \mathbf{u}_k^{(P)} = [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(P)}]\!] \\ &\text{subject to } u_{ik}^{(p)} > 0, i = 1, \ldots, M_p, k = 1, \ldots, K, p = 1, \ldots, P. \end{aligned} \tag{5.3}$$

where $\mathbf{u}_k^{(p)}$ is the kth column of the pth factor $\mathbf{U}^{(p)}$, and latent dimensionality K. Maximum likelihood learning of the tensor factors under a Gaussian noise model minimises the reconstruction error:

$$\min_{\{\boldsymbol{u}_k^{(p)}\}} \frac{1}{2} \Big\| \mathcal{X} - \sum_{k=1}^{K} \bigotimes_{p=1}^{P} \boldsymbol{u}_k^{(p)} \Big\|_F^2, \quad \text{s.t. } u_k^{(p)} \geq 0, \tag{5.4}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm, which is the sum of squares of all tensor elements. Two methods that achieve this are Positive Tensor Factorisation (PTF) (Welling and Weber, 2001) and non-negative tensor factorisation (NTF) (Shashua and Hazan, 2005). PTF learns parameters using multiplicative updates, which is an extension of the multiplicative-update learning that is used in non-negative matrix factorisation (example 2.5) to the case of tensor factors. NTF uses an EM-algorithm for learning.

5.2 A Bayesian Non-negative Tensor Factorisation

5.2.1 Model Construction

Consider a general non-negative tensor factorisation where the observed data is a P-way array, $\mathcal{X} \in \mathbb{R}^{M_1 \times \cdots \times M_P}$ and the dimensions of each mode are denoted by $M_1, \ldots, M_P$. Let $\mathcal{M} = \{1, \ldots, M_1\} \times \cdots \times \{1, \ldots, M_P\}$ be the index set over all elements in $\mathcal{X}$ and let $\mathbf{m} = (m_1, \ldots, m_P)$ be a P-tuple index in $\mathcal{M}$. For convenience, the total number of elements in $\mathcal{X}$ is denoted by $M = \prod_p M_p$.

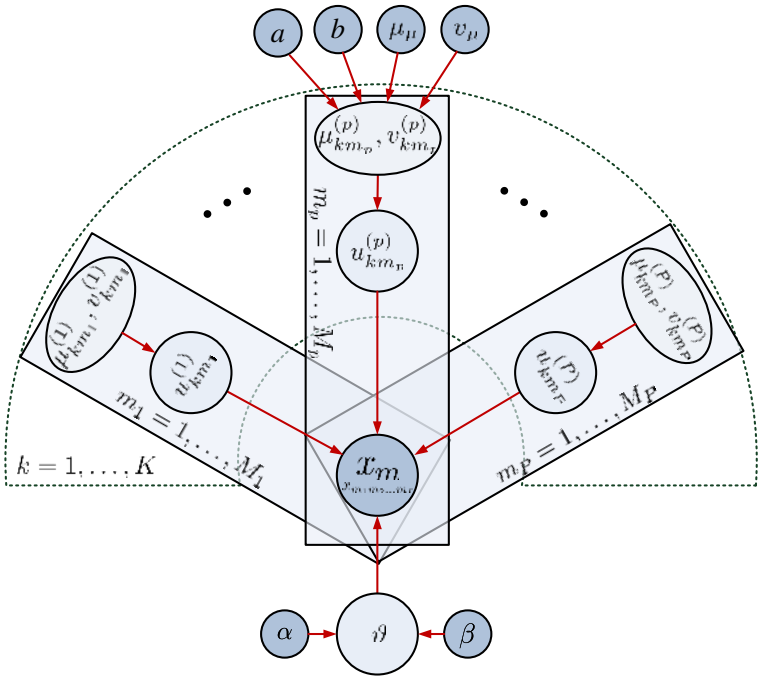


Figure 5.2: Graphical model of Bayesian NTF. The shaded node represents an observed variable, and the plates represent repeated variables.

We seek the following decomposition:

$$\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{k=1}^{K} \mathbf{u}_k^{(1)} \otimes \cdots \otimes \mathbf{u}_k^{(P)} = \sum_{k=1}^{K} \bigotimes_{p=1}^{P} \boldsymbol{u}_k^{(p)}, \tag{5.5}$$

with each element of $\hat{\mathcal{X}}$ computed as:

$$\hat{x}_{\boldsymbol{m}} = \sum_{k=1}^{K} \prod_{p=1}^{P} u_{km_p}^{(p)}. \tag{5.6}$$

This approximates $\mathcal{X}$ as the sum of K rank-1 tensors that are outer products of P non-negative vectors, $\mathbf{u}_k^{(p)} \in \mathbb{R}_+^{M_p}$. The elements of the non-negative vectors are $u_{km_p}^{(p)}$, where m_p is used as an index for co-ordinates of the P-tuple index $\mathbf{m}$. The vectors $\mathbf{u}_k^{(p)}$ are the tensor factors, and can be viewed as latent variables in a probabilistic setting. We use a latent variable modelling approach and describe our generative process for non-negative tensor data in figure 5.2.

We model an observed data point $x_{\mathbf{m}}$ as a Gaussian likelihood with mean $\hat{x}_{\boldsymbol{m}}$ given by the decomposition of equation (5.6) and variance ϑ,

$$p(x_{\boldsymbol{m}}|\{u_{km_p}^{(p)}\}, \vartheta) \quad = \quad \mathcal{N}(x_{\boldsymbol{m}}|\hat{x}_{\boldsymbol{m}}, \vartheta). \tag{5.7}$$

The prior on the data variance is an inverse Gamma distribution with shape and scale parameters α and β respectively,

$$p(\vartheta|\alpha,\beta) \quad = \quad \mathcal{G}^{-1}\left(\vartheta|\alpha,\beta\right) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\,\vartheta^{-(\alpha+1)}\exp\left(\frac{-\beta}{\vartheta}\right). \tag{5.8}$$

For each data point, we draw the corresponding latent variables $u^{(p)}_{km_p}$ from a rectified Gaussian distribution with unknown mean $\mu^{(p)}_{km_p}$ and variance $v^{(p)}_{km_p}$,

$$p\left(u^{(p)}_{km_p}\right) \quad = \quad \mathcal{N}^{R}\left(u^{(p)}_{km_p}|\mu^{(p)}_{km_p},v^{(p)}_{km_p}\right) \tag{5.9}$$

$$= \quad \frac{\sqrt{\frac{2}{\pi v^{(p)}_{km_p}}}}{\operatorname{erfc}\left(\frac{-\mu^{(p)}_{km_p}}{\sqrt{2v^{(p)}_{km_p}}}\right)}\exp\left(-\frac{(u^{(p)}_{km_p}-\mu^{(p)}_{km_p})^2}{2v^{(p)}_{km_p}}\right)H(u^{(p)}_{km_p}), \tag{5.10}$$

where $H(\cdot)$ is the Heaviside unit step function: $H(z)=1$ if $z>0, H(z)=0$ otherwise.

If the prior over $u^{(p)}_{km_p}$ had been a Gaussian, appropriate conjugate priors for the mean $v^{(p)}_{km_p}$ and the variance $v^{(p)}_{km_p}$ would be a Gaussian and inverse Gamma respectively. These priors are not conjugate to the rectified Gaussian and instead we choose a convenient joint conjugate prior density:

$$p(\mu^{(p)}_{km_p},v^{(p)}_{km_p}|\mu_\mu,v_\mu,a,b) = \frac{1}{c}\sqrt{v^{(p)}_{km_p}}\operatorname{erfc}\left(\frac{-\mu^{(p)}_{km_p}}{\sqrt{2v^{(p)}_{km_p}}}\right)\mathcal{N}(\mu^{(p)}_{km_p}|\mu_\mu,v_\mu)\,\mathcal{G}^{-1}\left(v^{(p)}_{km_p}|a,b\right),$$

where c is a normalisation constant. With this prior $\mu^{(p)}_{km_p}$ and $v^{(p)}_{km_p}$ decouple and the posterior conditional densities are Gaussian and inverse Gamma respectively.

We denote the set of unknown variables by $\mathbf{\Omega} = \left\{\{\mathbf{u}^{(p)}_k\},\vartheta,\{\mu^{(p)}_k\},\{\mathbf{v}^{(p)}_k\}\right\}$, and the set of hyperparameters $\mathbf{\Psi}=\{\alpha,\beta,a,b,\mu_\mu,v_\mu\}$. Following from the graphical model and equations (5.7) – (5.11) the joint probability is:

$$\begin{aligned}
p(\mathcal{X},\mathbf{\Omega}) \quad &= \quad p(\mathcal{X}|\{\mathbf{u}^{(p)}_k\},\vartheta)p(\vartheta|\alpha,\beta)p(\{\mathbf{u}_k\}|\{\boldsymbol{\mu}^{(p)}_k\},\{\boldsymbol{v}^{(p)}_k\})\times p(\{\mu^{(p)}_k\}|m_\mu,v_\mu)p(\{\boldsymbol{v}^{(p)}_k\}|a,b)\\
&\propto \quad \vartheta^{-\frac{M}{2}-\alpha-1}\prod_{\boldsymbol{m}\in\mathcal{M}}\exp\left\{\tfrac{1}{2\vartheta}\left(x_{\boldsymbol{m}}-\sum_{k=1}^{K}\prod_{p=1}^{P}u^{(p)}_{km_p}\right)^2\right\}\\
&\times \quad \exp\left(\tfrac{-\beta}{\vartheta}\right)\prod_{k=1}^{K}\prod_{p=1}^{P}\prod_{m_p=1}^{M_p}\left\{\exp\left(-\frac{(u^{(p)}_{km_p}-\mu^{(p)}_{km_p})^2}{2v^{(p)}_{km_p}}\right)H(u^{(p)}_{km_p})\right.\\
&\times \quad \left.\exp\left(-\frac{(\mu^{(p)}_{km_p}-\mu_\mu)^2}{2v_\mu}\right)(v^{(p)}_{km_p})^{-(\alpha+1)}\;\exp\left(\tfrac{-b}{v^{(p)}_{km_p}}\right)\right\}.
\end{aligned} \tag{5.11}$$

5.2.2 Model Properties

5.2.2.1 A Model for Bayesian NMF

The Bayesian NTF model we described in the previous section also provides a model for a Bayesian non-negative matrix factorisation (NMF). For a two-mode tensor, i.e. a matrix, the CP decomposition used in equation (5.5) reduces to a more familiar matrix decomposition with non-negatively constrained factors:

$$\hat{\mathcal{X}} \quad = \quad \mathbf{U}^{(1)}{\mathbf{U}^{(2)}}^{\top} \qquad \text{s.t. } \mathbf{U}^{(1)} \geq 0, \mathbf{U}^{(2)} \geq 0 \tag{5.12}$$

The specification presented here is based on a rectified Gaussian likelihood rather than the Poisson likelihood that is used in NMF (Lee and Seung, 1999) (described in example 2.5). In the limit as $\frac{\mu}{\sqrt{2v}} \to -\infty$, the rectified Gaussian $\mathcal{N}^R(x|\mu, v)$ becomes an Exponential distribution $\mathcal{E}(x| - \frac{\mu}{v})$, which is useful for inference in non-negative models using variational methods (Harva and Kaban, 2005), and shows the connections between this specification and the types of non-negative models based on the exponential distribution described previously (section 3.3.1; Seeger, 2008).

5.2.2.2 Permutation Indeterminacy

The NTF model has a permutation indeterminacy that must be taken into account when using the model for practical applications. The rank-one tensors which form the tensor product can be re-ordered arbitrarily, such that:

$$\mathcal{X} \quad = [\![\mathbf{U}_1, \ldots, \mathbf{U}_M]\!] \quad = [\![\mathbf{U}_1\mathbf{\Pi}, \ldots, \mathbf{U}_M\mathbf{\Pi}]\!] \tag{5.13}$$

for any $K \times K$ permutation matrix $\mathbf{\Pi}$.

The ordering of variables is irrelevant for parameter learning and for problems in prediction, since only the tensor reconstructions are considered and is identified under any permutation. But permutation is of concern when meaning is to be assigned to the latent factors. To ensure that this label switching is accounted for one of the approaches to handling this label switching, discussed previously in section 2.3.3, can be employed. A non-negative tensor factorisation has a number of useful properties over the unconstrained decomposition. In particular Lim and Comon (2009) show that a non-negative PARAFAC always has a solution, and study the conditions for this to hold. They consider several measures of proximity for the minimisation problem (5.5), showing that several norms as well as Bregman divergences result in the existence of an optimal solution.

5.2.3 Inference by MCMC

Posterior inference can be performed using MCMC techniques. We use two previously described MCMC methods for learning: Gibbs sampling and Hybrid Monte Carlo (HMC) sampling. Both methods were described in section 1.5.

Gibbs Sampling. The required full conditional distributions for Gibbs sampling can be obtained using the joint distribution of equation (5.11). The conjugate priors specified makes this process uncomplicated. We describe the Gibbs sampling steps in more detail here, since they were omitted in previous chapters. For each conditional posterior distribution, the posterior parameters are denoted by the same symbols as their prior parameters, with a bar. The notation $\boldsymbol{\Omega} \backslash u^{(p)}_{km_p}$ represents exclusion, where all parameters in the set $\boldsymbol{\Omega}$ are used except $u^{(p)}_{km_p}$.

The conditional distribution for $u^{(p)}_{km_p}$ is a rectified Gaussian:

$$p(u^{(p)}_{km_p}|\mathcal{X}, \boldsymbol{\Omega}\backslash u^{(p)}_{km_p}) = \mathcal{N}^R(u^{(p)}_{km_p}|\bar{\mu}^{(p)}_{km_p}, \bar{v}^{(p)}_{km_p}) \tag{5.14}$$

$$\bar{v}^{(p)}_{km_p} = \left(\frac{1}{\vartheta}\sum_{\boldsymbol{m}\in\mathcal{M}}\prod_{p'\neq p}\left(u^{(p')}_{km_{p'}}\right)^2 + \frac{1}{v^{(p')}_{km_{p'}}}\right)^{-1}$$

$$\bar{\mu}^{(p)}_{km_p} = \bar{v}^{(p)}_{km_p}\left\{\frac{1}{\vartheta}\sum_{\boldsymbol{m}\in\mathcal{M}}\left(\sum_{k'\neq k}\prod_{p} u^{(p)}_{km_p} - x_{\boldsymbol{m}}\right)\times\prod_{p'\neq p} u^{(p')}_{km_{p'}} + \frac{\mu^{(p)}_{km_p}}{v^{(p)}_{km_p}}\right\}. \tag{5.15}$$

The conditional posterior distribution of the data variance is an inverse Gamma distribution, $p(\vartheta|\mathbf{X}, \boldsymbol{\theta}\backslash\vartheta) = \mathcal{G}^{-1}(\vartheta|\bar{\alpha}, \bar{\beta})$, with shape and scale:

$$\bar{\alpha} = \alpha + \tfrac{M}{2}, \qquad \bar{\beta} = \beta + \tfrac{1}{2}\chi^2, \tag{5.16}$$

where $\chi^2 = \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|^2_F$ is the sum of squared errors. The conditional posterior distribution for $\mu^{(p)}_{km_p}$ is Gaussian, $p(\mu^{(p)}_{km_p}|\boldsymbol{X}, \boldsymbol{\theta}\backslash\mu^{(p)}_{km_p}) = \mathcal{N}(\mu^{(p)}_{km_p}|\bar{m}_\mu, \bar{v}_\mu)$, with variance and mean:

$$\bar{v}_\mu = \left(\frac{1}{v^{(p)}_{km_p}} + \frac{1}{v_\mu}\right)^{-1}, \qquad \bar{m}_\mu = \bar{v}\left(\frac{u^{(p)}_{km_p}}{v^{(p)}_{km_p}} + \frac{m_\mu}{v_\mu}\right). \tag{5.17}$$

The conditional posterior distribution for $v^{(p)}_{km_p}$ is an inverse Gamma, $p(v^{(p)}_{km_p}|\boldsymbol{X}, \boldsymbol{\theta}\backslash v^{(p)}_{km_p}) = \mathcal{G}^{-1}(v^{(p)}_{km_p}|\bar{a}, \bar{b})$, with shape and scale parameters:

$$\bar{a} = a + \frac{1}{2}, \qquad \bar{b} = b + \frac{1}{2}(u^{(p)}_{km_p} - \mu^{(p)}_{km_p})^2. \tag{5.18}$$

Hybrid Monte Carlo Sampling. We used HMC extensively in the previous chapters, and the application here follows the general approach previously described. For Bayesian NTF, the parameters $u^{(p)}_{km_p} \geq 0$, $\vartheta > 0$ and $v^{(p)}_{km_p} > 0$ can be transformed to unconstrained variables using the transformations: $u^{(p)}_{km_p} = \exp(\tilde{u}^{(p)}_{km_p})$, $\vartheta = \exp(\tilde{\vartheta})$, and $v^{(p)}_{km_p} = \exp(\tilde{v}^{(p)}_{km_p})$. After the inclusion of the Jacobian of the change of variables, the log joint probability obtained, which is the HMC Potential energy function, is:

$$\begin{aligned}\mathcal{L} &= \frac{1}{2\exp(\tilde{\vartheta})} \sum_{\boldsymbol{m}\in\mathcal{M}} \left(x_{\boldsymbol{m}} - \sum_{k=1}^{K} \prod_{p=1}^{P} \exp(\tilde{u}^{(p)}_{km_p}) \right)^2 + \left(\frac{M}{2} + \alpha \right) \tilde{\vartheta} + \frac{\beta}{\exp(\tilde{\vartheta})} \\ &+ \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{m_p=1}^{M_P} \left\{ \frac{\left(\exp(\tilde{u}^{(p)}_{km_p}) - \mu^{(p)}_{km_p} \right)^2 + 2b}{2\exp(\tilde{v}^{(p)}_{km_p})} + \frac{(\mu^{(p)}_{km_p} - \mu_\mu)^2}{2v_\mu} + a\tilde{v}^{(p)}_{km_p} - \tilde{u}^{(p)}_{km_p} \right\}.\end{aligned} \tag{5.19}$$

The derivatives required to complete the sampling procedure can be computed using the above joint probability for each of the unknown variables.

5.2.4 Experimental Performance

The Bayesian NTF model is evaluated using two data sets, using the testing methodology used throughout this thesis. The model is compared to the performance of the non-negative PARAFAC model of Bro and Jong (1999), which is able to deal with missing data using built-in EM iterations (Tomasi and Bro, 2002).

Synthetic Data. A synthetic 3-way data set was generated with three underlying factors resulting in a *50 × 50 × 3* tensor. The predictive performance using RMSE and NLP on held-out data is shown in figure 5.3. Both models perform well for small latent dimensionality K, but PARAFAC begins to overfit the data as highlighted by the trend towards zero training RMSE and increasing test RMSE. We examine the mixing properties of the sampler, which was run for 5 latent factors and 10,000 iterations, using the hairiness index $\hat{H}$ and the potential scale reduction factor $\hat{R}$ (see 1.5.4) for samples of the reconstruction product $[\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}]\!]$. Figure 5.4 shows representative hairiness plots for three parameters, a histogram of the hairiness index at the end of the chain for all parameters, and a histogram of the potential scale reduction factors obtained using 5 chains for all parameters (with dispersed starting points drawn a uniform distribution). The hairiness plots show good mixing of the parameters with 94% of the hairiness indices within the confidence bounds, and almost all parameters with an $\hat{R} < 1.1$, and together give no reason to suspect a lack of mixing in the Markov chain.

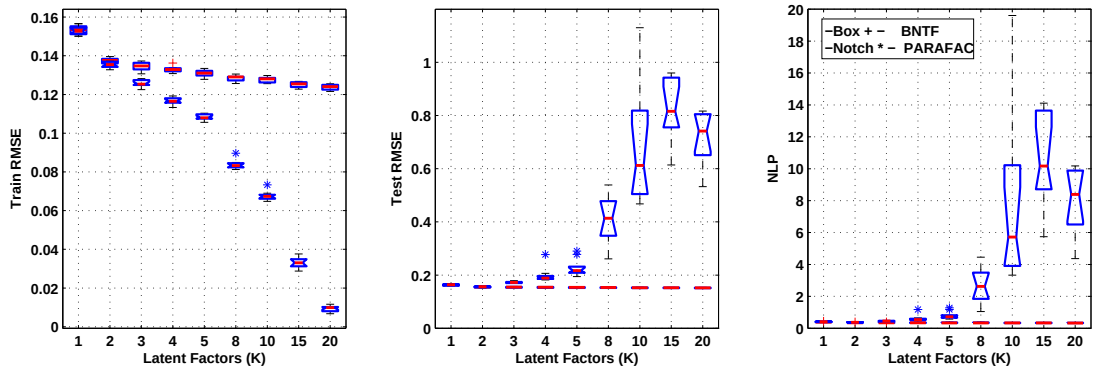


Figure 5.3: Performance of PARAFAC and Bayesian NTF using synthetic data for a varying number of latent factors.

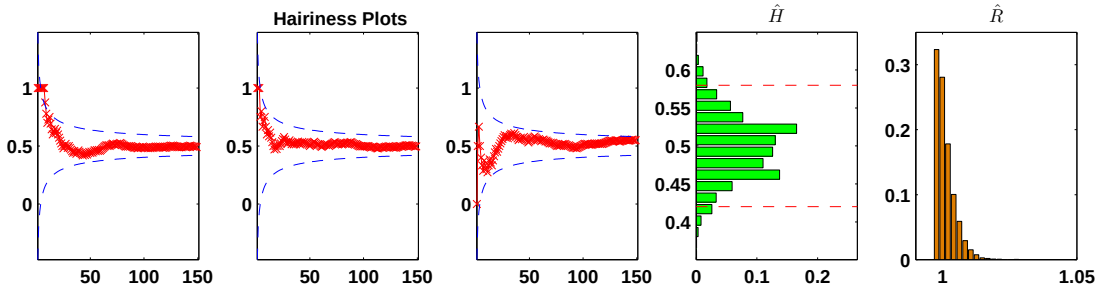


Figure 5.4: Analysis of mixing behaviour of the Bayesian non-negative tensor factorisation for the synthetic data. (a) - (c) Hairiness plots for 3 parameters. (d) Histogram of hairiness indices. (e) Histogram of PSRF values.

Colour of Beef. We use data from measurements collected in the study of colour changes in beef during storage under different conditions. The conditions are: storage time, temperature, time of light exposure and oxygen content, resulting in a 6-way tensor of *storage* $\times$ *temperature* $\times$ *oxygen* $\times$ *light* $\times$ *muscle* $\times$ *replicate*. The performance of both PARAFAC and the probabilistic NTF model in data reconstruction were compared, and are shown in figure 5.5. Non-negative PARAFAC predicts missing data well for model orders $K = 2$ and $K = 3$ in accordance with previous results on this data set (Bro and Jakobsen, 2002). For larger model orders however, PARAFAC tends to overfit the data. Our NTF model predicts missing data equally well or better at all model orders and does not overfit.

5.3 Amino Acid Fluorescence Application

Fluorescence spectroscopy is a technique used in the analysis of organic compounds and is used in the study of the properties and composition of compounds, in tracking bio-chemical reactions or in determining the conformal state of proteins. This analysis typically results in tensor data, since a number of samples are excited by light at a range of frequencies and the resultant emission spectra are recorded.

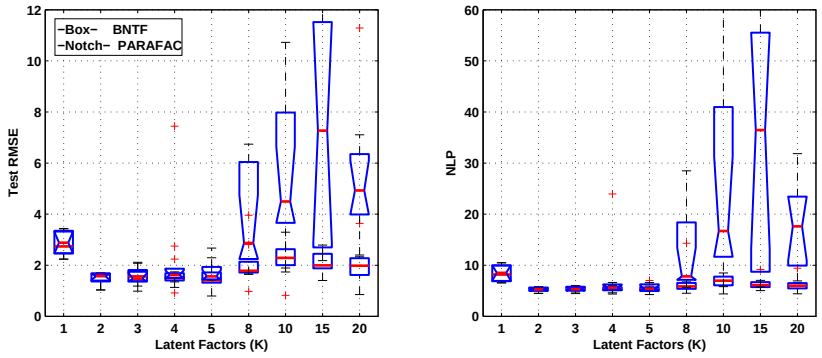


Figure 5.5: Performance of PARAFAC and Bayesian NTF for the colour of beef data for varying number of latent factors.

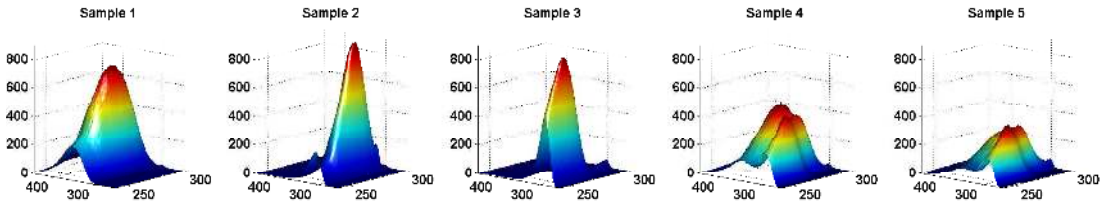


Figure 5.6: Fluorescence spectra of the five mixtures under study.

For this particular study, five organic solutions are analysed, each sample containing different amounts of the amino acids tyrosine, tryptophan and phenylalanine in a solvent. Each sample was excited at 61 wavelengths (240 – 300 nm in 1 nm intervals) and emission intensities are recorded at 201 wavelengths (250 - 450 nm in 1 nm intervals). This results in a tensor of *5 samples* $\times$ *61 excitation levels* $\times$ *201 emission levels*. Figure 5.6 shows the fluorescence spectra of the 5 samples. This can be seen a blind source separation problem, where data from a set of mixed sources is obtained and the task is to identify the constituent sources. Thus, with the fluorescence data from a set of solutions containing different proportions of amino acids, the task is to identify the individual amino acids in the solutions.

This data is inherently non-negative, making the use of our Bayesian NTF model applicable. Learning was performed with $K = 3$ latent factors. The three factors obtained are: $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}$ corresponding to the tensor modes for the samples, excitation wavelengths and emission wavelengths respectively. $\mathbf{U}^{(1)}$ gives insight into the proportion of the three amino acids used in each of the five samples. $\mathbf{U}^{(2)}$ gives insight into the absorption response of the three amino acids at the 61 excitation levels, and $\mathbf{U}^{(3)}$ gives insight regarding the fluorescence response of three amino acids at the 201 emission wavelengths. We run the Bayesian non-negative tensor factorisation model for 15,000 iterations using the first half as burn-in. We analyse the mixing of this chain as before, using the hairiness index $\hat{H}$ and potential scale reduction factor $\hat{R}$ on the reconstruction product $[\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}]\!]$, and is shown in figure 5.7. From this analysis, almost all parameters lie within the hairiness

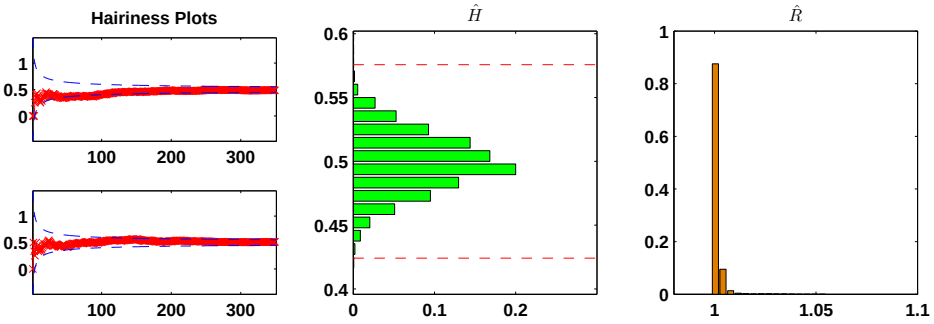


Figure 5.7: Mixing analysis for the amino acid fluorescence application. (a) Two representative hairiness plots. (b) Histogram of hairiness indices for all parameters. (c) Histogram of potential scale reduction factors for all parameters.

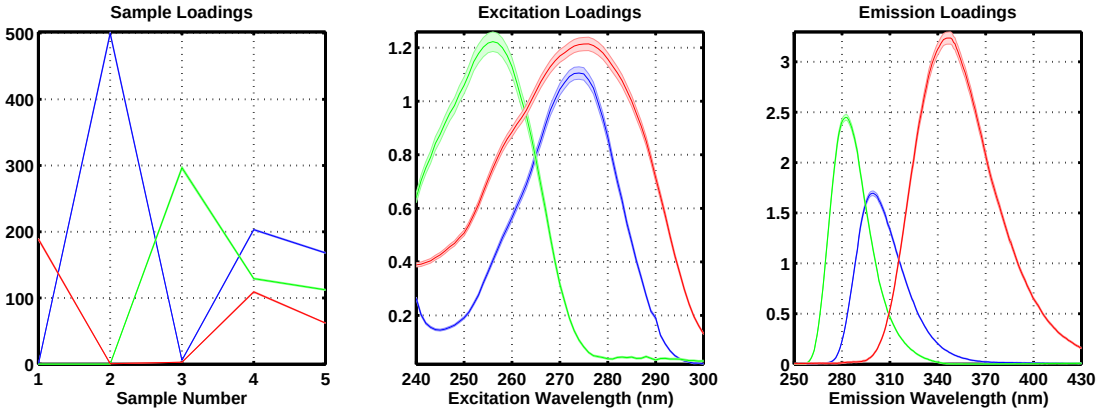


Figure 5.8: Plot of factor loadings obtained using the Bayesian NTF model for the amino acid fluorescence application, for the factor corresponding to (a) the tensor mode for the samples, (b) mode for the excitation wavelengths (c) mode for emission wavelengths. The colours indicate the three latent factors that were used and the error bars represent the variation in the coefficient when averaged over the samples obtained.

confidence bounds and have a PSRF less than 1.1, which give us no reason to suspect problems with mixing in the Markov chain.

The loadings obtained from the NTF analysis correspond to relative excitation or emission coefficients. Such a three-way data analysis cannot be done by multiple bilinear data analysis. We show the loading plot obtained for each of the latent factors in figure 5.8, which plots the coefficients in each of the the latent factors for each of the three latent dimensions (indicated by different colours). We observe an induced sparsity in the sample loadings, since all but one of the coefficients is non-zero in each of samples 1,2 and 3 (figure 5.8a). This sparsity aids interpretation and shows that the first three samples consist almost exclusively of one amino acid, whereas the last two samples are mixtures of all three.

Based on these plots, the biochemist examining these factors would be able to deduce what the constituent amino acids used in the 5 samples are. The red curve

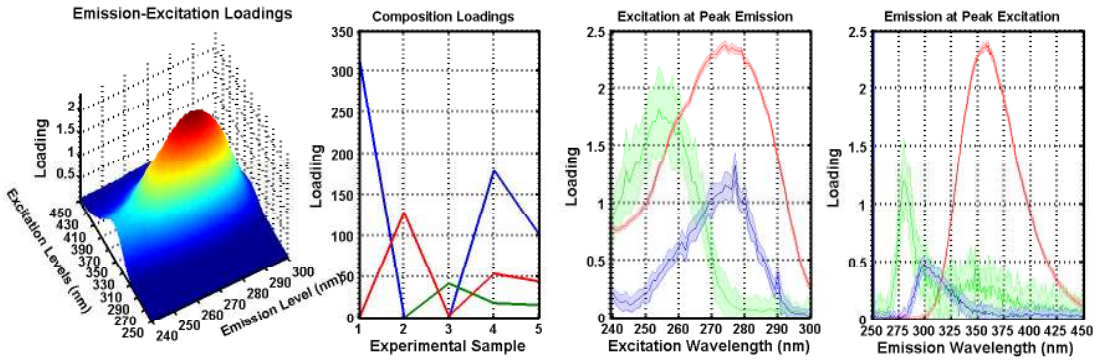


Figure 5.9: Amino acid data analysis using Bayesian NMF showing coefficients of the latent factors. (a) Emission-Excitation factor. (b) Coefficients for each of the five samples used. (c) The excitation profile at the corresponding peak emission in (a) averaged over samples over all excitation wavelengths considered. (d) The emission profile obtained at the peak excitation obtained in (a) averaged over samples over all emission wavelengths considered.

would be identified as Tyrosine since its emission and excitation profiles correspond to the known chemical properties for this amino acid, i.e peak excitation at 274 nm and emission at 303 nm. Similarly, the blue curve would be identified as Phenylalanine and the purple curve as Tryptophan (Berg et al., 2002, §3.1). This kind of analysis has become increasingly important since these techniques are used as diagnostics for the conformational states of proteins. In addition, the Bayesian approach gives a measure of the uncertainty of the emission or excitation coefficients, which gives insight into the environmental conditions of the solution. The small uncertainty region in these plots is indicative of the relative homogeneity of the solutions and minimal uncertainty in the peak excitation levels of the amino acids under study.

The importance of using a tensor based approach to modelling such naturally multi-way data can best be appreciated by considering the result and effort that would be needed using an equivalent matrix factorisation method, in this case a Bayesian non-negative matrix factorisation (NMF). The Bayesian NTF model is equivalent to a Bayesian NMF model when two-mode data is used, discussed in section 5.2.2.1. For NMF, the data must first be collapsed from a $P \times Q \times R$ tensor into a $P \times QR$ matrix, or collapsing by another mode - this process is referred to as matricisation. If K latent factors are to be estimated, matrix factorisation requires the computation of $N_{nmf} = (P + QR) \times K$ latent variables, whereas the corresponding tensor factorisation requires only $N_{ntf} = (P + Q + R) \times K$ latent variables. Since $K < P, Q, R$, this is a significant reduction in parameters to be estimated. For the amino acids study, the tensor approach requires only $1,325$ parameters compared to $36,798$ parameters for NMF.

We show the results produced by NMF in figure 5.9. These results are similar to those shown in figure 5.8 for NTF. To perform this analysis, we matricised the data to a 5×12261 matrix and assumed $K = 3$ latent factors. Using one sample chosen from the chain at convergence, the two matrix factors obtained are shown in figure 5.9(a),(b). Figure 5.9(a) shows only the first row of the 3×12261 factor matrix $\mathbf{U}_1$ and was reshaped to the 201×61 matrix to aid visualisation. Since the emission and excitation modes were flattened, NMF provides a joint emission-excitation factor. Figure 5.9(b) shows the composition loadings of the five samples and matches the corresponding factor produced by NTF. To obtain separate profiles for the emission and excitation behaviour, we search for the peak excitation mode and fix this when plotting the excitation behaviour, and similarly for the emission behaviour. These results are shown in figure 5.9(c),(d), which match the general trends observed in the NTF results, but are very noisy. The tensor model allowed a direct way of visualising these factors, provided smoother factors and required less effort both computationally and in visualising the results, and is preferable for the analysis of such naturally 3-way data.

5.4 Discussion

Recently, Chu and Ghahramani (2009) presented the probabilistic Tucker model, which is a generative specification of the Tucker model, as opposed to the generative description of the CP model we considered here. In addition to the advantages that a probabilistic CP model gives: allowing missing data to be handled easily and accounting for uncertainty in the observed data, a probabilistic Tucker model provides a basis upon which other models can be built and generalised. Since the CP model is a special case of the Tucker model, this model provides an important general purpose model for the analysis of tensor data. The probabilistic Tucker model embeds the Tucker decomposition in a linear Gaussian framework for estimation, which allows the core tensor to be integrated out and learning is achieved by MAP estimation using gradient descent. Bayesian learning is not widely considered in the literature for tensor decompositions. We developed one of the first fully Bayesian models for non-negative tensor factorisation in Schmidt and Mohamed (2009), and have expanded on this work here. A fully Bayesian exploration of the probabilistic Tucker model and the improvements that this may bring is thus one avenue of further development.

A probabilistic CP or Tucker model can also be used as the basis upon which tensor models generalised to the exponential family can be developed, using the approach discussed in chapter 2. In such a setting, marginalisation of the core tensor will not be possible, resulting in a harder inference problem. Inference in tensor models other than MAP/ML or MCMC have also not been considered and variational inference approaches may be one interesting avenue of exploration.

5.5 Tensor Factorisation in Context

The origin of tensor factorisations can be traced to problems in linear algebra, in particular to the work of Hitchcock (1927), followed by models for multi-way data by Cattell (1944). Tensor decompositions truly came into being after the work of Tucker (1966) in developing the Tucker model and by Harshman (1970) with the PARAFAC model. Interestingly, these models were developed in psychometrics, the same field from which Factor Analysis was borne. Tensor decompositions soon became established as a powerful method for analysis in chemometrics, with the work of Bro (Bro and Jong, 1999; Bro and Jakobsen, 2002) quite prominent. There are many variants of both Tucker and CP models and these are well described in the review papers by Kolda and Bader (2007) and Acar and Yener (2009).

Non-negative tensor factorisation came to prominence following the popularity of non-negative matrix factorisation (Paatero and Tapper, 1994; Lee and Seung, 1999). Welling and Weber (2001) discuss a positive matrix factorisation that uses multiplicative updates similar to those used for NMF by Lee and Seung (1999). Cichocki et al. (2007) present an NTF algorithm based on minimising alpha and beta divergences, and Shashua and Hazan (2005) present an EM algorithm for problems in computer vision. Hazan et al. (2005) show in a related paper that the use of tensor approaches for image analysis is better suited to handling spatial redundancy than using NMF with vectorised images.

We developed one of the first fully Bayesian approaches to learning in non-negative tensor factorisations using MCMC methods in Schmidt and Mohamed (2009) and in this chapter. Porteous et al. (2008) present models for parametric and non-parametric Bayesian tensor factorisation. The parametric model is constructed by considering P interacting LDA models (Blei et al., 2003) for each of the P tensor modes. A non-parametric version is obtained by considering the non-parametric analogue of the LDA model, which is the Hierarchical Dirichlet process (HDP). The model is briefly described, but no evaluation or other discussion is presented, requiring further work in this area. Other recent developments in probabilistic modelling for tensors include the probabilistic Tucker models of Chu and Ghahramani (2009) who discuss an EM algorithm for learning. A Bayesian tensor model has also been used for modelling relational data (Sutskever et al., 2009), which can be seen as Bayesian adaptation of the Tucker 2 modelling approach (Acar and Yener, 2009), using a prior specification based on the Chinese Restaurant Process.

5.6 Summary

In this chapter we developed generalisations of latent factor models to multi-way or tensor data sets. Models for tensor data allow the natural relationships in the data to be maintained and provide a means of learning concise descriptions of data. We focussed on the construction of models for non-negative tensor decompositions based on the CP-decomposition framework, and described approaches to fully Bayesian learning. Our model decomposes an observed n-way data set into n factor matrices, which represent an underlying description for each of the tensor modes. We demonstrated the robustness of the Bayesian approach and highlighted the importance and practicality of non-negative representations of data with an application in fluorescence spectroscopy.

We discussed the relationship between the non-negative tensor factorisation described in this chapter and the more familiar non-negative matrix factorisation, along with the wide set of existing work in this area. A natural extension of these ideas, is the construction of tensor models generalised to the exponential family. This can also be coupled with an investigation of alternative approximate inference schemes. The applicability of tensor models has thus far been restricted to applications in chemometrics, though this is a diverse area of applied science. The application to research fields such as those in relational learning are still open, and there remains much unexplored territory for the application of tensor models.

Chapter 6

Discussion and Conclusion

Models for matrix factorisation have become an essential tool in a wide variety of research areas, whether this be in online rental settings for movie recommendation, in research environments for the analysis of gene expression data, or in food science in deciding the storage and transport conditions for various foods. In this thesis, we have made a number of advances in probabilistic models for matrix and tensor factorisation that allow us to apply these methods to the new and diverse application areas that are increasingly found. We recap our contributions here.

We developed a Bayesian model for matrix factorisation in chapter 2 that is generalised to the exponential family, which allows modelling of data that may be counts, binary, non-negative or a heterogeneous set of these data types. This unifying model for unsupervised learning plays a complementary role to the generalised linear models for regression. We have also shown that a number of popular models in use are special cases of the generalised latent variable model. Our results showed that the Bayesian exponential family PCA model produces better results than a corresponding maximum likelihood approach, avoids overfitting and produces useful predictions in a number of settings. We also developed a new post-processing strategy for dealing with factor identifiability, allowing samples to be generated from which meaningful averages can be computed. The Scotch data analysis showed a typical application in marketing, the use of the generalised model in the analysis of binary purchasing data and the insight obtained after post-processing samples for identifiability.

In chapter 3 we developed both optimisation and Bayesian approaches to learning latent representations with sparsity. We extended the exponential family framework by specifying a generic loss function and optimisation algorithm that allowed generalised learning with sparsity using the L_1 norm. We extended the exponential family framework in a second way by developing new sparse Bayesian latent variable

models and novel inference procedures, considering both scale-mixture priors and discrete mixture priors. Importantly, we provided the first comparison of sparse learning using these three approaches: optimisation using the L_1 norm, sparse Bayesian learning with continuous sparsity-favouring priors and spike-and-slab priors. Our results show that the spike-and-slab has the best performance on held out data on all data sets and produces accurate reconstructions even with restricted running times. We demonstrated these methods in a diverse set of applications including text modelling, robot planning and psychology showing the flexibility of the sparse models developed.

In chapter 4 we developed a novel and simple approach for Binary PCA based on dichotomising underlying Gaussian variables. We derived fully the equations that match moments between correlated binary variables and latent Gaussian variables, and demonstrated its application with a simple example. The algorithm allows for sampling of correlated binary variables with desired means and covariances and gives insight into the implications of dichotomising a Gaussian distribution. We subsequently developed a binary PCA algorithm that combines Gaussian dichotomisation with existing approaches for computing principal components, resulting in a new binary PCA algorithm.

We showed that latent variable modelling techniques can be naturally extended to data sets that are represented as a multi-dimensional arrays or tensors in chapter 5. We developed the first fully Bayesian non-negative tensor factorisation model and described its properties and two MCMC sampling algorithms. We demonstrated the effectiveness of our model in a fluorescence spectroscopy application, where maintaining the tensor structure of the data coupled with Bayesian inference led to cleaner and more interpretable results.

A number of significant themes have underpinned the development of this thesis. Our overarching theme has been that of Bayesian statistical approaches to latent variable modelling and inference. The Bayesian approach emphasised accounting for uncertainty and averaging over model parameters, rather than searching for a single parameter setting. This Bayesian thinking established intuitive approaches to model development based on the specification of hierarchical Bayesian models, which were used to specify generative processes of data. Bayesian inference overcame problems with data over-fitting and the limitations of maximum likelihood methods, allowed for a straightforward approach to dealing with missing data, and in all the settings considered gave better predictive performance than maximum likelihood approaches.

A second theme, has been that of unification. We developed a unification of various models for unsupervised learning in the exponential family PCA method

described in chapter 2, using the shared properties of the exponential family of distributions. We showed that this model encompasses a number of existing models including non-negative matrix factorisation (Paatero and Tapper, 1994; Lee and Seung, 1999), probabilistic latent semantic analysis (Hofmann, 1999) and logistic PCA (Tipping, 1999; Schein et al., 2003). Similarly, the generic loss function we described in chapter 3 provided a unification of various approaches to sparse modelling, such as the Lasso (Tibshirani, 1996) and regularised logistic regression (Lee et al., 2006b; Schmidt et al., 2007). The unification of many continuous sparsity-favouring priors was also described based on the scale-mixture of Gaussians, and provides a means with which to reason and explore the various continuous sparse priors available. Finally, we showed the unification of matrix and tensor factorisation approaches in chapter 5.

A third theme has been the consideration of structure in data. One advantage of Bayesian methods is the ability to include prior information into the model specification, and it is through this prior specification that structure in the data can be explored. Wide classes of models and structural assumptions are spanned by considering various priors for the latent representation, such as factor analysis (chapter 2; Spearman, 1904; Bartholomew and Knott, 1999), mixture models (Newcomb, 1886; Titterington et al., 1985), partial membership models (Heller et al., 2008) and latent feature selection (chapter 3); see table 1.1. We also considered correlation in data (chapter 4), and learning non-negative representations of data (chapter 5).

We have shown that latent variable models are applicable to a wide range of applications. Emphasis was placed throughout this thesis on exploratory analysis and the visualisation of data using the inferred latent representation. We addressed problems such as collaborative filtering, recommendation and advertising using latent variable models for the task of missing data imputation and data reconstruction (chapter 2). We explored data from psychological experiments and used our new models to obtain insight into the factors contributing to decision making (chapter 3). We also explored the application of factor models in monitoring food quality (chapter 5). The analysis of binary data is also a theme carried throughout, with coverage of various approaches to handling such data either by Gaussian dichotomisation using the probit construction (chapter 4), or based on Bernoulli likelihoods giving the logit representation (chapter 2).

Sampling based approaches to learning have also featured strongly. Gibbs sampling is restricted to conjugate cases where full conditional distributions can be derived (chapter 5), whereas the auxiliary variable samplers used are much more widely applicable (chapters 1, 2, 3). Both Hybrid Monte Carlo and slice sampling required the evaluation of the joint probability to be implemented successfully, and

are well suited to the non-conjugate models explored in this thesis. Other approaches to inference are also of relevance though they have not been explored here. These include alternative approaches based on moment-matching, such as that discussed in chapter 4, and other approximate inference such as variational approximations.

Sparsity also presents a number of interesting future research directions. A full theoretical understanding of the priors obtained using the scale-mixture of Gaussians construction is needed, as well as new methods for encoding structure in the sparse representations that are learnt. Aspects of sparsity in the non-parametric Bayesian setting can be further explored and would allow for highly flexible models to be developed. Tensor models have promise for the analysis of relational data, where relationships between entities are naturally represented by a data tensor. The concluding comments of each chapter made mention of the various other opportunities for future work.

In this thesis, we have contributed to the methodology, learning and applications of latent variable models. We described generalisation in its widest sense: generalisation of the types of data that are modelled, generalisation of the types of priors that are used and the generalisation of the data structures that are considered. We developed models for various facets of this generalisation. We described new exponential family generalisations that allow for the analysis of many diverse types of data. Sparsity in latent representations was explored to allow for latent factor selection, and we extended the number of latent factors used to model tensor data. The models presented are flexible enough to allow useful predictions to be made, provide insight into the underlying structure of the data, have highlighted the inherent relationships between models popular in the day-to-day analysis of data; and demonstrate the practical use and advantages of Bayesian methods in unsupervised statistical settings.

The more than a century-long history of latent variable models is evidence of the indelible impact this class of models has had on statistics. Despite their long history, research into latent variable models remains active, and this thesis has made a number of contributions in advancing this important area of research.

References

F. Abramovich, V. Grinshtein, and M. Pensky. On optimality of Bayesian testimation in the normal means problem. *Annals of Statistics*, 35(5):2261–2286, 2007. (page 77)

E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering*, 21:6–20, 2009. (pages 94 and 107)

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008. (page 79)

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993. (page 90)

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):99–102, 1974. (page 58)

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725., 2009. (page 64)

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003. (page 12)

C. Archambeau, N. Delannay, and M. Verleysen. Robust probabilistic projections. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 33–40, 2006. (page 52)

K. S. Azoury and M. K. Warmuth. Relative Loss Bounds for On-Line Density Estimation with the Exponential Family of Distributions. *Machine Learning*, 43(3):211–246, 2001. (page 8)

F. Bach. Structured sparsity-inducing norms through submodular functions. In *Neural Information Processing Systems (NIPS)*, 2010. (page 74)

R. R. Bahadur. *Studies in Item Analysis and Prediction*, volume VI of *Stanford Mathematical Studies in the Social Sciences*, chapter "A representation of the joint distribution of responses to n dichotomous items", pages 158 – 168. Stanford University Press, 1961. (page 90)

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005. (page 8)

O. Barndorff-Nielsen. Hyperbolic distributions and distributions on hyperbolae. *Scandinavian Journal of Statistics*, 5(3):151–157, 1978. (page 59)

D. J. Bartholomew and M. Knott. *Latent variable models and factor analysis*, volume 7 of Kendall's library of statistics. Arnold, 2nd edition, 1999. (pages 23 and 111)

M. S. Bartlett. Multivariate analysis. *Supplement to the Journal of the Royal Statistical Society*, 9(2):pp. 176–197, 1947. URL http://www.jstor.org/stable/2984113. (page 28)

M. J. Beal. *Variational Algorithms for approximate Bayesian inference*. PhD thesis, University of Cambridge, 2003. (page 48)

M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003. (page 64)

J. M. Berg, J. L. Tymoczko, and L. Stryer. *Biochemistry*. W. H. Freeman and Co., 5th edition, 2002. (page 105)

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, 1st edition, 1994. (page 12)

A. Beskos, N. Pillai, G. Roberts, J.-M. Sanz-Serna, and A. Stuart. Optimal tuning of hybrid Monte Carlo. http://arxiv.org/abs/1001.4460, 2010. (page 16)

M. Bethge and P. Berens. Near-maximum entropy models for binary neural resprentations of natural images. In *Neural Information Processing Systems (NIPS)*, volume 21, 2007. (page 92)

P. J. Bickel and K. A. Doksum. *Mathematical statistics: Basic ideas and selected topics*, volume I. 2001. (pages 37 and 38)

C. M. Bishop. Bayesian PCA. In *Neural Information Processing Systems (NIPS)*, pages 382–388, 1999. (pages 36 and 51)

C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, August 2006. (pages 12, 25, 32, and 33)

D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, 2005. (pages 81, 82, and 92)

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993 – 1022, 2003. (pages 51 and 107)

T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5–6):629–654, 2008. (page 73)

S. Boyde and L. Vandenberge. *Convex Optimization*. Cambridge University Press, 2004. (page 31)

L. M. Bregman. 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 Physics*, 7:200 – 217, 1967. (page 8)

L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37: 373–384, November 1995. (page 72)

N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88(421):9–25, 1993. (page 27)

R. Bro and M. Jakobsen. Exploring complex interactions in designed data using GEMANOVA. *Journal of Chemometrics*, 16(6):294–304, May 2002. (pages 102 and 107)

R. Bro and S. D. Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1999. (pages 101 and 107)

J. Brodie, I. Daubechies, C. D. Mol, D. Giannone, and I. Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Science*, 106(30):12267–12272, 2009. (pages 54 and 78)

S. P. Brooks and G. O. Roberts. Convergence assessment techniques for Markov Chain Monte Carlo. *Statistics and Computing*, 8:319 – 335, 1998. (pages 19 and 20)

W. Buntine and A. Jakulin. Discrete components analysis. In *Subspace, Latent Structure and Feature Selection*, volume 3940/2006, pages 1–33. Springer (LNCS), 2006. (page 51)

E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006. (pages 56, 74, and 78)

F. Caron and A. Doucet. Sparse Bayesian nonparametric regression. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*. 2008. (pages 76 and 79)

C. Carvalho, N. Polson, and J. G. Scott. The Horseshoe estimator for sparse signals. *Biometrika*, 97(2):465 – 480, 2010a. (page 57)

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008. (pages 53, 54, 65, 78, and 79)

C. M. Carvalho, H. Lopes, and O. Aguilar. Dynamic stock selection strategies: a structured factor model framework. In *Bayesian Statistics 9*, 2010b. (pages 54 and 78)

O. Caster, G. Noren, D. Madigan, and A. Bate. Large-scale regression-based pattern discovery: The example of screening the WHO global drug safety database. In *Proceedings of the KDD Workshop on Mining Medical Data*, 2008. (page 54)

R. Cattell. "Parallel proportional profiles" and other principles for determining the choice of factors by rotation. *Psychometrika*, 9(4):267–283, December 1944. (page 107)

N. R. Chaganty and H. Joe. Range of correlation matrices for dependent Bernoulli random variables. *Biometrika*, 93(1):197 – 206, 2006. (pages 86 and 92)

R. Chartrand. Exact reconstructions of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14:707–710, 2007. (page 73)

S. Chib and E. Greenberg. Analysis of multivariate probit models. *Biometrika*, 85(2): 347–361, 1998. (pages 90 and 91)

S. B. Choy and J. S. Chan. Scale mixtures distributions in statistical modelling. *Australian & New Zealand Journal of Statistics*, 50(2):135–146, 2008. (page 58)

W. Chu and Z. Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009. (pages 106 and 107)

A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. Amari. Non-negative tensor factorization using alpha and beta divergences. In *IEEE conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages III–1393 – III–1396, 2007. (page 107)

J. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38:826–844, 1973. (page 78)

D. Clarkson. Estimating the standard errors of rotated factor loadings by jackknifing. *Psychometrika*, 44(3):297–314, September 1979. (page 43)

M. Collins, S. Dasgupta, and R. Schapire. A generalization of PCA to the exponential family. In *Advances in Neural Information Processing (NIPS)*, volume 14, pages 617 – 624. 2002. (pages 28, 30, 31, 51, and 67)

P. Common. Independent component analysi – a new concept? *Signal Processing*, 36: 287 – 314, 1994. (page 79)

M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91:883–904, 1996. (page 20)

D. R. Cox and N. Wermuth. On some models for multivariate binary variables parallel in complexity with the multivariate Gaussian distribution. *Biometrika*, 89(2):462–469, 2002. (pages 84 and 92)

I. Csiszár and G. Tusnády. Information geometry and alternating minimisation procedures. *Statistics and Decisions, Supplement Issue*, 1:205–237, 1984. (page 31)

J. P. Cunningham. *Algorithms for understanding motor cortical processing and neural prosthetic systems. Chapter 4: Calculating Multivariate Gaussian Probabilities*. PhD thesis, Stanford University, 2009. (page 87)

P. Damien, J. Wakefield, and S. Walker. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of The Royal Statistical Society Series B*, 61(2):331–344, 1999. (page 18)

A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Neural Information Processing Systems (NIPS)*, volume 17, pages 41–48, 2005. (page 78)

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977. (page 31)

L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986. URL http://cg.scs.carleton.ca/~luc/rnbookindex.html. (page 18)

D. L. Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications in Pure and Applied Mathematics*, 59:797–829, 2004. (page 56)

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4): 1289–1306, April 2006. (pages 56, 74, and 78)

D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM Journal of Applied Mathematics*, 49(3):906–931, 1989. (page 78)

F. Doshi-Velez and Z. Ghahramani. Correlated non-parametric latent feature models. In *Uncertainty in Artificial Intelligence (UAI)*, 2009. (pages 82 and 92)

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216 – 222, 1987. (page 14)

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, December 2009. (page 78)

D. Dueck and B. Frey. Probabilistic sparse matrix factorization. Technical Report PSI-2004-23, University of Toronto, 2004. (pages 54 and 79)

Y. D. Edwards and G. M. Allenby. Multivariate analysis of multiple response data. *Journal of Marketing Research*, 4:321 – 324, 2003. (pages xi, 46, and 47)

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004. (page 78)

L. El Ghaoui and A. Gueye. A convex upper bound on the log-partition function for binary graphical models. In *Advances in Neural Information Processing Systems*, 2008. (page 49)

L. J. Emrich and M. R. Peidmonte. A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304, 1991. (pages 82 and 92)

L. Fahrmeir and G. Tutz. *Multivariate statistical modelling based on generalized linear models*. Springer, 2001. (page 27)

M. A. Figueiredo and S. Member. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1150–1159, 2003. (page 79)

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302 – 332, 2007. (page 31)

A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990. (pages 12 and 13)

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, 2nd edition, 2004. (pages 19, 20, 32, 33, 40, 41, and 65)

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984. (page 13)

A. Genz. Numerical computation of multivariate Normal probabilities. *Journal of Computational and graphical statistics*, 1:141–149, 1992. (page 87)

E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. (page 78)

J. F. Geweke and G. Zhou. Measuring the pricing error of the arbitrage pricing theory. *Review of Financial Studies*, 9:557 – 587, 1996. (page 36)

Z. Ghahramani and M. Beal. Variational inference for bayesian mixture of factor analysers. In *Advances in Neural Information Processing Systems*. 2000. (page 49)

S. Ghosal and A. Roy. *Perspective in Mathematical Sciences I*, chapter "Bayesian nonparametric approach to multiple testing", pages 139–164. World Scientific Publishing Company, 2009. (page 77)

W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics (Chapman & Hall/CRC Interdisciplinary Statistics)*. Chapman and Hall/CRC, 1995. (pages 12, 20, and 64)

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996. (page 35)

G. J. Gordon. Generalized^2 linear^2 models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 577–584, 2002. (page 27)

J. Griffin and P. Brown. Alternative prior distributions for variable selection with very many more variables than observations. Technical report, University of Warwick, 2005. (page 79)

J. E. Griffin and P. J. Brown. Inference with Normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188, 2010. (page 78)

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian Buffet Process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16. NIPS, 2006. (pages 66 and 76)

C. Guihenneuc-Jouyaux and J. Rousseau. Laplace expansions in markov chain monte carlo algorithms. *Journal of Computational and Graphical Statistics*, 14(1):pp. 75–94, 2005. (page 64)

Y. Guo. Supervised exponential family principal component analysis via convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 569–576, 2008. (page 52)

R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):84, 1970. (pages 94 and 107)

M. Harva and A. Kaban. A variational Bayesian method for rectified factor analysis. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 185–190, 2005. (page 99)

T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. London: Chapman & Hall, 1990. (page 27)

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970. (page 12)

K. Hayashi, J. Hirayama, and S. Ishii. Dynamic exponential family matrix factorization. In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09)*, 2009. (page 52)

T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3D non-negative tensor factorization. In *International Conference of Computer Vision (ICCV)*, pages 50–57, 2005. (page 107)

K. Heller, S. Williamson, and Z. Ghahramani. Statistical models for partial membership. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008. (pages 52 and 111)

G. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. (pages 82 and 92)

F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6:164–189, 1927. (page 107)

T. Hofmann. Probabilistic latent semantic indexing. In *22nd Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57, 1999. (pages 30, 51, and 111)

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933. (page 23)

J. Huang, S. Ma, and C.-H. Zhang. Adaptive lasso for sparse high dimensional regression models. *Statistica Sinica*, 18:1603–1618, 2008. (page 53)

H. Ishwaran and J. S. Rao. Detecting differentially expressed genes in microarrays using Bayesian model selection. *Journal of the American Statistical Association*, 98: 438–455, 2003. (page 53)

H. Ishwaran and J. S. Rao. Spike and Slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005. (pages 57, 59, 60, and 78)

S. Ji, Y. Xue, , and L. Carin. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 58(6):2346 – 2356, 2008. (page 75)

I. M. Johnstone and B. W. Silverman. Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Annals of Statistics*, 32(4):1594–1694, 2004. (pages 65 and 77)

I. Joliffe. *Principal Components Analysis*. Springer, 2nd edition, 2002. (pages 2, 23, and 29)

C. Jutten and J. Hérault. Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1 – 10, 1991. (page 79)

A. Kaban and E. Bingham. ICA-based binary feature construction. In *Proceedings of the 6th International Conference on Independent Component Analysis and Signal Separation (ICA)*, volume 6, LNCS 3889, pages 140 – 148, 2006. (page 67)

A. Kaban and R. J. Durrant. Learning with $l_{q<1}$ vs l_1-norm regularisation with exponentially many irrelevant features. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 580–596, 2008. (page 73)

C. Kemp and J. B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Science*, 105(31):10687–10692, 5 August 2008. (pages 67, 70, and 71)

M. E. Khan, B. Marlin, G. Bouchard, and K. Murphy. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2010. (page 49)

J. Kittler and J. Föglein. Contextual classification of multispectral pixel data. *Image and Vision Compution*, 2:13 – 29, 1984. (page 14)

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation (ICA)*, volume 7, 2007. (pages 76, 77, and 78)

D. Knowles and Z. Ghahramani. Nonparametric bayesian sparse factor models with application to gene expression modelling. *Annals of Applied Statistics*, In Press, 2010. (page 76)

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. Technical Report SAND2007-6702, Sandia National Laboratories, California, 2007. (pages 94 and 107)

T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009. (page 67)

J. Lafferty and L. Wasserman. Rodeo: Sparse, greedy nonparametric regression. *Annals of Statistics*, 36(1):28–63, 2008. (page 73)

D. Lambert. Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992. (page 2)

L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253–1278, 2000. (page 95)

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(21):788 –791, 1999. (pages 29, 51, 96, 99, 107, and 111)

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 801–808, 2007. (page 31)

H. Lee, R. Raina, A. Teichman, and A. Y. Ng. Exponential family sparse coding with applications to self-taught learning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1113–1119, 2009. (pages 52 and 79)

S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1 regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2006a. (page 78)

S. Lee, H. Lee, P. Abbeel, and A. Ng. Efficient L1 regularized logistic regression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2006b. (pages 57, 78, and 111)

F. Leisch, A. Weingessel, and K. Hornik. On the generation of correlated artificial binary data. Technical report, Vienna University of Economics and Business Administration, 1998. (pages 84, 86, and 92)

M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 1998. (page 75)

W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international Conference on Machine learning (ICML)*, pages 577–584, 2006. (pages 82 and 92)

L.-H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics*, 23:432 – 441, 2009. (page 99)

H. F. Lopes and M. West. Bayesian model assessment in factor analysis. *Statistica Sinica*, 14:41 – 67, 2004. (page 36)

H. F. Lopes, D. Gamerman, and E. Salazar. Generalized spatial dynamic factor models. *Computational Statistics & Data Analysis*, In Press, Corrected Proof, 2010. (page 52)

M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. (page 73)

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, June 2003. (pages 12, 32, and 64)

J. H. Macke, P. Berens, A. S. Eker, A. S. Tolias, and M. Bethge. Generating spike trains with specified correlation coefficients. *Neural Computation*, 21:397–423, 2009. (pages 84, 86, 91, and 92)

D. Madigan, P. Ryan, S. Simpson, and I. Zorych. Bayesian methods in pharmacovigilance. In *Bayesian Statistics 9*, 2010. (page 54)

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010. (page 31)

J. Marchini, B. Howie, S. Myers, G. McVean, and P. Donnelly. A new multipoint method for genome-wide association studies via imputation of genotypes. *Nature Genetics*, 39:906–913, 2007. `https://mathgen.stats.ox.ac.uk/impute/impute.html`. (page 70)

B. M. Marlin, M. Schmidt, and K. P. Murphy. Group sparse priors for covariance estimation. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009. (page 74)

S. Martino, R. Akerkar, and H. Rue. Approximate bayesian inference for survival models. Technical report, Norwegian University of Science and Technology, 2010. (page 50)

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006. (page 78)

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. (page 12)

T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001. (page 87)

T. J. Mitchell and J. J. Beauchamp. Variable selection in linear regression (with discussion). *Journal of the American Statistical Association*, 83:1023–1036, 1988. (pages 59 and 78)

S. Mohamed, K. Heller, and Z. Ghahramani. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems (NIPS)*, 2009. (pages 51 and 67)

I. Moustaki and M. Knott. Generalized latent trait models. *Psychometrika*, 65(3):391–411, 2000. (pages 28, 30, 31, 37, and 51)

I. Murray, R. P. Adams, and D. J. MacKay. Elliptical slice sampling. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 541–548, 2010. (page 19)

R. M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993. (pages 12, 14, 15, and 20)

R. M. Neal. Markov chain Monte Carlo methods based on 'slicing' the density function. Technical Report 9722, Department of Statistics, University of Toronto, 1997. (page 18)

R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3):705 767, 2003. (pages 18 and 64)

R. M. Neal. *Handbook of Markov Chain Monte Carlo*, chapter "MCMC using Hamiltonian dynamics". Chapman & Hall / CRC Press, 2010. (pages 14, 15, and 18)

J. Nelder and R. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A*, 135(3):370–384, 1972. (pages 25 and 26)

Netflix. Netflix prize, 2009. URL http://www.netflixprize.com. (pages 47 and 81)

S. Newcomb. A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, 8:343–366, 1886. (pages 23 and 111)

NIST. Digital library of mathematical functions, March 2010. URL http://dlmf.nist.gov/. (page 59)

R. B. O'Hara and M. J. Sillanpää. A review of Bayesian variable selections methods: What, how and which. *Bayesian Analysis*, 4(1):85 – 118, 2009. (pages 60 and 78)

S. D. Oman and D. M. Zucker. Modelling and generating correlated binary variables. *Biometrika*, 88:287 – 290, 2001. (page 92)

P. Orbanz and Y. W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*. Springer, 2010. (page 76)

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. (pages 51, 107, and 111)

M. Y. Park and T. Hastie. L_1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69: 659–677, 2007. (page 78)

T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008. (page 79)

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559 – 572, 1901. (pages 23 and 50)

K. Pearson. On a new method of determining correlation between a measured character A, and a character B, of which only the percentage of cases wherein B exceeds (or falls short of) a given intensity is recorded for each grade of A. *Biometrika*, 7 (1/2):pp. 96–105, 1909. (pages 83 and 91)

N. G. Polson and J. G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. In *Bayesian Statistics*, volume 9. 2010. (pages 76 and 79)

I. Porteous, E. Bart, and M. Welling. Multi-HDP: A nonparametric Bayesian model for tensor factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2008. (page 107)

B. F. Qaqish. A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, 90(2): 455–463, 2003. (pages 82 and 92)

R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984. (page 43)

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004. (pages 12 and 20)

S. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing (NIPS)*, volume 10, pages 626–632. 1998. (pages 24 and 51)

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999. (page 23)

N. Roy and G. Gordon. Exponential family PCA for belief compression in POMDPs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1043–1049, 2003. (page 52)

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009. (pages 49 and 50)

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 25, 2008. (pages 33, 52, and 68)

F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986. (page 78)

A. I. Schein, L. K. Saul, and L. H. Ungar. A generalized linear model for principal components analysis of binary data. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2003. (pages 29, 51, and 111)

M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for L1 regularization: A comparative study and two new approaches. In *In Proceedings of the European Conference on Machine Learning (ECML)*. 2007. (pages 57, 78, and 111)

M. N. Schmidt and S. Mohamed. Probabilistic non-negative tensor factorization using Markov chain Monte Carlo. In *Proceedings ofn the 17th European Signal Processing Conference (EUSIPCO)*, 2009. (pages 106 and 107)

J. G. Scott and J. O. Berger. An exploration of aspects of Bayesian multiple testing. *Journal of Statistical Planning and Inference*, 136(7):2144 – 2162, 2006. (pages 70 and 77)

M. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008. (pages 75 and 99)

M. Seeger, S. Gerwinn, and M. Bethge. Bayesian inference for sparse generalized linear models. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2007. (pages 62 and 79)

A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, pages 792–799, 2005. (pages 96 and 107)

A. P. Singh. *Efficient Models for Relational Learning*. PhD thesis, Machine Learning Department, Carnegie Mellon University, 2009. (page 32)

J. Skilling and D. J. C. MacKay. Slice sampling – a binary implementation. *Annals of Statistics*, 31(3):753–755, June 2003. Discussion of *Slice Sampling* by Radford M. Neal. (page 18)

C. Spearman. "General intelligence," objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904. (pages 23, 50, and 111)

N. Srebro and T. Jaakkola. Sparse matrix factorization for analyzing gene expression patterns. In *NIPS Workshop on Machine Learning Techniques for Bioinformatics*, 2001. (pages 54, 78, and 79)

N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorizations. In *Advances in Neural Information Processing Systems (NIPS)*, 2005a. (page 79)

N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorizations. In *Advances in Neural Information Processing Systems (NIPS) 17*, 2005b. (page 52)

M. Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society. Series B*, 62(4):795 – 809, 2000. (page 44)

D. Stern, R. Herbrich, and T. Graepel. Matchbox: Large scale Bayesian recommendations. In *Proceedings of the 18th International World Wide Web Conference*, 2009. (page 48)

G. W. Steward. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551 – 566, 1993. (page 50)

I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1821–1828, 2009. (page 107)

R. H. Swendsen and J. S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987. (page 14)

R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996. (pages 55, 56, 57, 78, and 111)

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, 67(1):91–108, 2005. (pages 73 and 78)

M. E. Tipping. Probabilistic visualisation of high dimensional binary data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 11, pages 592 – 598, 1999. (pages 29, 40, 51, and 111)

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001. (page 73)

M. E. Tipping and C. M. Bishop. Probabilistic principal components analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, 1997. (pages 24, 31, 36, and 51)

D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, 1985. (pages 23 and 111)

G. Tomasi and R. Bro. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75:163–180, 2002. (page 101)

L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31 (3):279–311, September 1966. (pages 94 and 107)

UCI Data. UCI machine learning repository SPECT heart data set. `http://archive.ics.uci.edu/ml/datasets/`. (pages 43, 67, and 68)

S. A. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009. (page 72)

M. Wainwright, T. Jaakkola, and A. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313 – 2335, 2005. (page 49)

M. J. Wainwright and M. I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 54 (6-1):2099–2109, 2006. (page 9)

Y. Weiss, H. S. Chang, and W. T. Freeman. Learning compressed sensing. In *Snowbird*. 2007. (page 75)

M. Welling and M. Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22 (12):1255 – 1261, 2001. (pages 96 and 107)

M. Welling, C. Chemudugunta, and N. Sutter. Deterministic latent variable models and their pitfalls. In *SIAM Conference on Data Mining (SDM)*, pages 196 – 207, 2008. (pages 32 and 51)

M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987. (page 58)

M. West. Bayesian factor regression models in the "large p, small n" paradigm. In *Bayesian Statistics 7*, pages 723–732, 2003. (pages 65 and 79)

J. Yoon, S. Wilson, K. Kayabol, and E. E. Kuruoglu. Variant functional approximations for latent gaussian models. Technical report, Trinity College Dublin, Dept of Statistics, 2010. (page 50)

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006. (pages 73 and 78)

R. Zass and A. Shashua. Nonnegative sparse PCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1561–1568, 2006. (pages 31, 54, and 79)

A. Zellner. An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association*, 57(298):pp. 348–368, 1962. (page 92)

H. T. Zhu, J. C. Eickhoff, and P. Yan. Generalized linear latent variable models for repeated measures of spatially correlated multivariate data. *Biometrics*, 61:674 – 683, 2005. (page 52)

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. doi: 10.1198/016214506000000735. URL http://pubs.amstat.org/doi/abs/10.1198/016214506000000735. (page 72)

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. Technical report, Technical report, statistics department, Stanford University, 2004. (pages 54 and 78)