Modelling inference strategies and robust clustering topologies

Adam Farooq
Doctorate of Philosophy

Aston University
June, 2022

Abstract

Latent variable models are used extensively in unsupervised learning within the Bayesian paradigm, these include (but are not limited to) mixture models which can be used for clustering, and linear Gaussian models which can be used for dimensionality reduction. Clustering aims to find some underlying groups within a data set, where data points that belong to the same group (also called a cluster) are more 'similar' to one another than data points that belong to different groups. Dimensionality reduction aims to reduce the dimension of a data set while minimising some information loss, for example, if two data points are relatively 'close' to one another in the observed data space, then they should also be relatively 'close' in the reduced dimension data space.

The Bayesian paradigm offers rules for learning from data and integrating out uncertainty, however, it can be a curse within latent variable models. For example, any misspecification of the likelihood within a mixture model will result in incorrect clustering. To combat this, we propose novel techniques to assist latent variable models to learn meaningful information.

We first propose a mixture model for clustering and density estimation of count data, which unlike other mixture models from the exponential family of distributions does not make a strong a-priori assumption on the dispersion of the observed data. The proposed model uses a mixture of *Panjer* distributions, which learns the dispersion of the observed data in a data-driven manner; we call this the Panjer mixture model. We study practical inference with the Panjer mixture model and propose an efficient maximisation-maximisation scheme for training the Panjer mixture model and demonstrate its utility on different data sets.

We propose an approach that aims to robustify the likelihood of a model with respect to any likelihood misspecification. Unlike the vast existing work, the proposed model is not an attempt to infer the parameters of a model in a robust manner, but it aims to learn the correct data-generating distribution. This is done by using pseudo-points in the data space which have an empirical density that is 'close' to the true data generating density; this is done using a statistical distance called the maximum mean discrepancy, which compares the summary statistic(s) between two distributions using the reproducing kernel Hilbert space (RKHS). The proposed model is applied to mixture models where each component is represented using pseudo-points. The advantage of the proposed mixture model is demonstrated on a variety of data sets.

We also propose two discrete-continuous latent feature models which can be used for dimensionality reduction to assist in tasks such as exploratory analysis, pre-processing, data visualisation, and related tasks. A constrained feature allocation prior is placed on the discrete component of the proposed models; we call these the *adaptive factor analysis* model and the *adaptive probabilistic principal component analysis model.* We also derive efficient inference schemes for each model. The usefulness of the proposed models is demonstrated in tasks such as feature learning, data visualisation, and data whitening using different data sets.

Bayesian nonparametric priors assume that the parameter space is infinite, this allows for flexible modelling, for example, the Dirichlet process (DP) can be used in mixture models to learn the number of clusters in a data-driven manner. However, the existing discrete Bayesian nonparametric priors assume that the latent space is discrete. We propose two novel discrete Bayesian nonparametric priors which generalise existing Bayesian nonparametric priors such as the *beta-Bernoulli* process, we call these the *discrete marked beta-binomial* process, and the *marked beta-negative-binomial* process. Furthermore, marginal processes for special cases of the proposed processes have also been derived which allow for efficient sampling; we call this the *multi-scoop Indian buffet process* and the *infinite-scoop Indian buffet process.*

Acknowledgements

First and foremost, I'd like to thank my supervisor Dr. Yordan P. Raykov, as cliché as it sounds his continuous support has been the sole reason for completing this thesis. His enthusiasm and mentorship over the past five years have guided me through the ups and downs of research. I particularly enjoyed our thought-provoking evening meetings. I am also grateful to Prof. Max Little for the countless opportunities he gave me. I would also like to thank Prof. Magnus Rattray for giving me the opportunity for the research visit in his group at the University of Manchester; and Dr. Mudassar Iqbal for his assistance in applying machine learning methods.

I would also like to thank the entire faculty of the mathematics department at Aston University; especially Prof. David Saad, Prof. Helen Higson, and Dr. Sotos Generalis for their support and input on my research.

Last but not least, I would like to thank my entire family, without their continuous support this work would have been impossible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unsupervised learning is a type of learning which discovers *patterns* from unlabelled data; where the term pattern is used loosely as it depends on the modelling goal. In this thesis, we primarily focus on two different techniques of unsupervised learning. The first technique we focus on is clustering which in summary can be described as finding some underlying groups of a data set, where data points that belong to the same group (or cluster) are more 'similar' to one another than data points that belong to different groups. An example of clustering can be grouping customers (also called segmenting) by their spending habits [1]. The second technique this thesis focuses on is dimensionality reduction which can be summarised as reducing the dimension of a data set while minimising some 'information loss'; for example, one may want to maximise the variance from the observed data set and its dimensional reduced counterpart. This has many applications such as data visualisation, for example visualising (for exploratory data analysis) 784-dimensional data in a two-dimensional plot [2].

These methods can be applied in several ways, for example, the K-means clustering algorithm [3] which is the most popular clustering algorithm uses a deterministic approach that minimises the distance between data points and cluster centroids. These methods can also be applied with methods from *Bayesian* statistics which proved a number of added benefits such as incorporating prior knowledge. Many of the unsupervised learning problems are solved using *latent variable models* which assume that there exist some unobserved (latent) variables that generated the observed data points; for example, these latent variables can represent clusters.

However, these latent variable models suffer from assumptions made about the model. For example, the *factor analysis* model assumes that factors are constructed using all data points, this can often lead to learning *global structure* over *local structure* [4]. In this thesis, we propose to enhance existing latent variable models to extract more meaningful information from data.

1.1 Contributions

This thesis makes the following contributions:

- We derive a generalised mixture model for clustering and density estimation of count data: *Panjer mixture model* which is generalisation of models such as *binomial mixture model*, *Poisson mixture model* and *negative-binomial mixture model* etc. The additional benefit of the proposed model is that it makes no assumption about the dispersion of the data, which results in better density estimation and by extension clustering.

- We propose an approach that aims to robustify a model with regards to any likelihood misspecification: *maximum mean discrepancy pseudo-point marginal* which uses pseudo-points in the data space to represent the likelihood using the maximum mean discrepancy (MMD). The MMD is used to evaluate the likelihood of an observation given the pseudo-points by comparing the summary statistic(s) between the two using the reproducing kernel Hilbert space (RKHS). The proposed model is applied to mixture models, where each mixture component has its own set of pseudo-points. Experiential results suggest that the proposed model works well even if the mixture components overlap in the Euclidean space.

- We derive two discrete-continuous latent feature models which can be used for exploratory analysis, pre-processing, data visualisation, and related tasks: *adaptive factor analysis* model and the *adaptive probabilistic principal component analysis* *model.* Existing methods tend to employ beta-Bernoulli priors which couple the feature frequency with the portion of total variance which leads to having a small number of data points being represented by all features; and a large number of data points being represented by a small number of features [5]. We propose an alternative approach that allows for better control over the feature to data point allocation. This new approach is based on the *multivariate hypergeometric* distribution which is a two-parameter discrete distribution that decouples feature sparsity and dictionary size, hence capturing both common and rare features in a parsimonious way.

- We propose two Bayesian nonparameteric processes: the *discrete marked beta-binomial process* & its marginal the *multi scoop Indian buffet process* for efficient sampling, and the *marked beta-negative-binomial process* & its marginal the *infinite scoop Indian buffet process.* Both of these models extend the latent space to consist of counts, this work builds upon the works done by [6, 7] and extends the marginal process called the Indian buffet process proposed in [8].

1.2 Thesis organisation

This thesis is organised in the following way:

- In Chapter 2 we give a broad overview of Bayesian modelling and probabilistic mixture models. We also review two of the most common priors used in discrete Bayesian nonparametrics: *Dirichlet process* and the *beta-Bernoulli process.* These will be used throughout this thesis.

- In Chapter 3 we derive the generalised mixture model for clustering and density estimation of count data. We also propose two different schemes to infer the parameters: (1) *Expectation-maximisation* (EM), and (2) *Maximisation-maximisation* (MM). Finally, the proposed mixture model is compared with three other mixture models using three different data sets using two criteria: (1) Density estimation, and (2) Clustering accuracy.

- In Chapter 4 we first investigate the problems one can face if the likelihood is misspecified. We then derive an approach that can robustify a likelihood to any likelihood to misspecification; we do this by representing the likelihood using pseudo-points via MMD which is then applied to a mixture model setup. We demonstrate the superior clustering of the proposed model when compared to existing methods which aim to solve the same misspecification problem.

- In Chapter 5 we investigate how exiting discrete-continuous latent feature models fail to capture the underlying feature allocation due to their priors. We then propose two novel discrete-continuous latent feature models which decouple feature sparsity and dictionary size. A Gibbs and EM scheme is proposed to infer all the parameters of the models; this includes an efficient (and practical) approach to infer over the Stiefel manifold. We demonstrate how well the proposed models work in different applications; this includes visualising hand-written digit images and discovering brain activity from fMRI data.

- In Chapter 6 we extend the existing works done in Bayesian nonparameteric by proposing two novel processes that do not confine the latent space to be binary; where the discrete marked beta-binomial process assumes that the latent space is discrete but has an upper bound (like the binomial distribution), and the marked beta-negative-binomial process assumes that the latent space is discreet but have no upper bound (like the negative-binomial distribution). We discuss how sampling from the process can be strenuous, and therefore show how the marginal processes offer a simpler alternative to sampling from their respective processes.

- In Chapter 7 we summaries all the previous chapters, and discuss the future directions of the work presented in this thesis.

Chapter 2

Bayesian nonparametrics

This chapter will introduce concepts that will be used in the other chapters of this thesis. This chapter does not aim to give a detailed induction, but it aims to give a broad introduction to various concepts such as Bayesian statistics.

2.1 Bayesian

Bayesian statistics is designed to integrate over uncertainty. The cornerstone of the Bayesian framework is Bayes' theorem [9]:

$$posterior \propto prior \times likelihood,$$

where the *prior* encapsulates any beliefs about a phenomenon in the absence of any data, the *likelihood* describes the probability of observing data, and the *posterior* describes the updated beliefs about a phenomenon given the data. Generally speaking, θ represents some unobserved parameter within a model that describes a phenomenon, then given some observed data one can *learn* θ in a coherent way.

More formally let $\{y_n\}_{n=1}^N \in \mathcal{Y}$ be N observed data points in some data space $\mathcal{Y}$ that have been sampled from some distribution parameterised by θ. In the Bayesian paradigm the unobserved parameter θ is treated as a random variable, which given some data has a posterior distribution that is a product of two probability distributions: (1)

The prior distribution $\mathrm{P}\left(\theta\right)$ which encapsulate any beliefs about θ in the absence of any data and (2) The likelihood $\mathrm{P}\left\{y_n\}_{n=1}^{N}|\theta\right)$ of observing N data points $\{y_n\}_{n=1}^{N}$ given θ. The posterior distribution of θ is:

$$\mathrm{P}\left(\theta|\{y_n\}_{n=1}^{N}\right) = \frac{\mathrm{P}\left(\theta\right)\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\theta\right)}{\mathrm{P}\left(\{y_n\}_{n=1}^{N}\right)}, \tag{2.1}$$

where $\mathrm{P}\left(\{y_n\}_{n=1}^{N}\right)$ is the normalising constant (also called the marginal distribution).

2.1.1 Exponential family of distributions

The exponential family of distributions [10] is a family of distributions which have a *probability mass function* (PMF) or *probability density function* (PDF) take the following form:

$$\mathrm{P}\left(y|\theta\right) = \exp\left(\eta\left(\theta\right)T\left(y\right) - A\left(\eta\left(\theta\right)\right) + C\left(y\right)\right), \tag{2.2}$$

where θ is the distributional parameter, $\eta(\cdot)$ is the natural parameter, $T(\cdot)$ is the sufficient statistic, $A\left(\cdot\right)$ is the log-partition and $C\left(\cdot\right)$ is the log-base measure. It should be noted that equation (2.2) only shows the univariate case.

The Bernoulli distribution is one example of the exponential family of distributions. Let $y \in \{0, 1\}$ be a draw from a Bernoulli distribution with success probability $p \in [0, 1]$, then:

$$\begin{aligned}
y|p \sim & \mathrm{Bernoulli}\left(p\right), \\
\mathrm{P}\left(y|p\right) = & p^{y}\left(1-p\right)^{1-y} \\
= & \exp\left(y\ln\left(\frac{p}{1-p}\right) + \ln\left(1-p\right)\right) \\
= & \exp\left(\eta\left(\theta\right)T\left(y\right) - A\left(\eta\left(\theta\right)\right) + C\left(y\right)\right),
\end{aligned}$$

where the distributional parameter $\theta = p$, the natural parameter $\eta(\theta) = \ln\left(\frac{p}{1-p}\right)$, the sufficient statistic $T(y) = y$, the log-partition $A\left(\eta(\theta)\right) = -\ln\left(1-p\right) = -\ln\left(1+\exp\left(\theta\right)\right)$, and the log-base measure $C\left(y\right) = 0$.

Conjugacy

Computing the posterior distribution $\mathrm{P}\left(\theta|\{y_n\}_{n=1}^N\right)$ is often difficult as the normalising constant from equation (2.1) $\mathrm{P}\left(\{y_n\}_{n=1}^N\right) = \int \mathrm{P}\left(\{y_n\}_{n=1}^N|\theta\right)\mathrm{P}\left(\theta\right) d\theta$ is often not available in closed form. This can be avoided if the prior and likelihood distributions both have the same functional form with respect to the distributional parameter θ; this is called *conjugacy* which is guaranteed for the exponential family of distributions.

Beta-Bernoulli: The beta distribution is the conjugate prior to the 'success' parameter in the Bernoulli distribution. Let $p \in [0,1]$ be the 'success' parameter of the Bernoulli distribution. The beta distribution is placed as a prior on the parameter p:

$$\begin{aligned} p|\alpha,\beta &\sim \mathrm{Beta}\left(\alpha,\beta\right), \\ \mathrm{P}\left(p|\alpha,\beta\right) &= \frac{p^{\alpha-1}\left(1-p\right)^{\beta-1}}{\mathrm{B}\left(\alpha,\beta\right)}, \end{aligned}$$

where $\mathrm{B}\left(\alpha,\beta\right) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ is the beta function, $\Gamma\left(\cdot\right)$ is the Gamma function, and $\{\alpha,\beta\}$ are some hyper-parameters which encapsulate any belief on p in the absence of any data. The observations $\{y_n\}_{n=1}^N$ are independently and identically (i.i.d.) sampled from a Bernoulli distribution, which would result in the likelihood:

$$\begin{aligned} \{y_n\}_{n=1}^N|p &\overset{i.i.d.}{\sim} \mathrm{Bernoulli}\left(p\right), \\ \mathrm{P}\left(\{y_n\}_{n=1}^N|p\right) &= \prod_{n=1}^N p^{y_n}\left(1-p\right)^{1-y_n} \\ &\propto p^{\sum_n y_n}\left(1-p\right)^{N-\sum_{n=1}^N y_n}. \end{aligned}$$

Which results in the posterior distribution:

$$\begin{aligned} \mathrm{P}\left(p|\{y_n\}_{n=1}^N,\alpha,\beta\right) &\propto \mathrm{P}\left(p|\alpha,\beta\right) \times \mathrm{P}\left(\{y_n\}_{n=1}^N|p\right) \\ &\propto p^{\alpha-1}\left(1-p\right)^{\beta-1} \times p^{\sum_{n=1}^N y_n}\left(1-p\right)^{N-\sum_{n=1}^N y_n} \\ &\propto p^{\alpha+\sum_{n=1}^N y_n-1}\left(1-p\right)^{\beta+N-\sum_{n=1}^N y_n-1}, \\ p|\{y_n\}_{n=1}^N,\alpha,\beta &\sim \mathrm{Beta}\left(\alpha+\sum_{n=1}^N y_n, \beta+N-\sum_{n=1}^N y_n\right), \end{aligned}$$

which is still a beta distribution.

Gaussian-Gamma-Gaussian: The Gaussian-Gamma[1] [11] distribution is the conjugate prior to the mean and precision (inverse of the variance) parameter in the Gaussian distribution. Let μ and λ be the mean and precision parameters of the Gaussian distribution. The Gaussian-Gamma distribution is placed as a prior on the parameters μ and λ:

$$\begin{aligned}
\mu, \gamma | \mu_0, \lambda_0, \alpha_0, \beta_0 &\sim \mathcal{NG}\left(\mu_0, \lambda_0, \alpha_0, \beta_0\right), \\
\lambda | \alpha_0, \beta_0 &\sim \mathcal{G}\left(\alpha_0, \beta_0\right), \\
\mu | \mu_0, \lambda_0, \lambda &\sim \mathcal{N}\left(\mu_0, \left(\lambda_0 \lambda\right)^{-1}\right), \\
\mathrm{P}\left(\mu, \lambda | \mu_0, \lambda_0, \alpha_0, \beta_0\right) &= \frac{\beta_0^{\alpha_0}\sqrt{\lambda_0}}{\sqrt{2\pi}\Gamma\left(\alpha_0\right)} \lambda^{\alpha_0 - \frac{1}{2}} \exp -\frac{1}{2}\left(\lambda_0\left(\mu-\mu_0\right)^2 + 2\beta_0\right),
\end{aligned}$$

where $\mathcal{G}(\cdot)$ is the Gamma distribution, $\mathcal{N}(\cdot)$ is the Gaussian distribution, and $\{\mu_0, \lambda_0, \alpha_0, \beta_0\}$ are some hyper-parameters which encapsulate any belief on μ and λ in the absence of any data. The observations $\{y_n\}_{n=1}^N$ are i.i.d. samples from a Gaussian distribution, which would result in the likelihood:

$$\begin{aligned}
\{y_n\}_{n=1}^N | \mu, \lambda &\overset{i.i.d.}{\sim} \mathcal{N}\left(\mu, \lambda^{-1}\right) \\
\mathrm{P}\left(\{y_n\}_{n=1}^N | \mu, \lambda^{-1}\right) &\propto \prod_{n=1}^N \sqrt{\frac{\lambda}{2\pi}} \exp\left(-\frac{\lambda}{2}\left(y_n - \mu\right)^2\right) \\
&= \left(\frac{\lambda}{2\pi}\right)^{N/2} \exp\left(-\frac{\lambda}{2}\sum_{n=1}^N \left(y_n - \mu\right)^2\right),
\end{aligned}$$

which results in the posterior distribution:

$$\begin{aligned}
\mathrm{P}\left(\mu, \lambda | \{y_n\}_{n=1}^N, \mu_0, \lambda_0, \alpha_0, \beta_0\right) \propto &\mathrm{P}\left(\mu, \lambda | \mu_0, \lambda_0, \alpha_0, \beta_0\right) \times \mathrm{P}\left(\{y_n\}_{n=1}^N | \mu, \lambda\right) \\
\propto &\sqrt{\lambda} \exp\left(-\frac{\lambda_0 \lambda \left(\mu - \mu_0\right)^2}{2}\right) \lambda^{\alpha_0 - 1} \exp\left(-\beta_0 \lambda\right) \\
&\times \lambda^{\frac{N}{2}} \exp\left(-\frac{\lambda}{2}\sum_n^N \left(y_n - \mu\right)^2\right),
\end{aligned}$$

[1] Also called the Normal-Gamma distribution

which results in:

$$
\begin{aligned}
\mu, \lambda | \{y_n\}_{n=1}^{N}, \mu_0, \lambda_0, \alpha_0, \beta_0 \sim & \mathcal{NG}\left(\mu_N, \lambda_N, \alpha_N, \beta_N\right), \\
\mu_N = & \frac{\lambda_0 \mu_0 + N\bar{y}}{\lambda_0 + N}, \\
\lambda_N = & \lambda_0 + N, \\
\alpha_N = & \alpha_0 + \frac{N}{2}, \\
\beta_N = & \beta_0 + \frac{1}{2}\sum_{n=1}^{N}\left(y_n - \bar{y}\right)^2 + \frac{\lambda_0 N \left(\bar{y} - \mu_0\right)^2}{2\left(\lambda_0 + N\right)}, \\
\bar{y} = & \frac{1}{N}\sum_{n=1}^{N} y_n,
\end{aligned}
$$

which is still a Gaussian-Gamma distribution; see [12] for the full derivation.

2.1.2 Finite mixture models

Mixture models provide an important framework in decomposing complex multi-modal distributions into a mixture of *simpler* distributions. A common application of this is clustering, where each mixture component represents a distinct cluster [13]; the total number of which are fixed to K. Each component has some mixing weight π_k, this weight is often interpreted as the prior probability of an observation being generated by the kth component, which is sampled from the Dirichlet distribution:

$$\pi_1, \ldots, \pi_K | \alpha_1, \ldots, \alpha_K \sim \text{Dirichlet}\left(\alpha_1, \ldots, \alpha_K\right),$$

where $\{\alpha_k\}_{k=1}^{K}$ are some positive hyper-parameters. The parameters of each k component is sampled form some distribution H:

$$\{\theta_k\}_{k=1}^{K} \sim H.$$

If conjugacy is desired, then H is chosen depending on the likelihood of the data. Each observation y_n (where $n \in \{1, \ldots, N\}$) is independently and identically generated in the following way:

$$\begin{aligned} c_n \sim & \text{Categorical}\left(\pi_1, \dots, \pi_N\right), \\ y_n | c_n, \theta_{c_n} \sim & \mathcal{F}_{\theta_{c_n}}, \end{aligned}$$

where $c_n \in \{1, \dots, K\}$ indicates which one of the K mixture component generated the nth observation and $\mathcal{F}_\theta$ is some distribution parameterised by θ. Setting H to be a conjugate prior to the likelihood $\mathcal{F}_{(.)}$ is a natural choice within the Bayesian paradigm. For example, in a mixture of univariate Gaussian's, $\mathcal{F}_{\theta_k}$ would be a Gaussian distribution, $\theta_k = \{\mu_k, \lambda_k\}$ would be a collection of mean and precision (inverse of the variance) parameters and a natural choice for H would be the Gaussian-Gamma distribution.

The whole mixture model is redistricted to K mixture components; K can be known using some a-prior knowledge or using other diagnostics such as Bayesian information criterion [14]. However, fixing K may become problematic when more data is observed as it may require the number of mixture components to grow. This motivates us to propose a more flexible set of models which grow in *complexity*[2] as more data is observed, this is known as *Bayesian nonparameterics.*

2.2 Bayesian nonparameterics

From the previous section, we can see that the traditional Bayesian paradigm treats the parameter space as finite, hence the prior (and the subsequent posterior) are defined on some finite space. The *Bayesian nonparametrics* (BNP) paradigm treats the parameter space as infinite, which requires one to use a prior (and subsequent posterior) that is defined on some infinite-dimensional space; which is the path of a stochastic process. Two of the most common discrete stochastic processes used in BNP are the *Dirichlet process* (DP) [15] and the *Beta process* [16]; both of which are covered in the sections below.

[2] In terms of the number of parameters which is a result of more number of components.

2.2.1 Dirchlet process

Definition

The *gamma process* is a positive Lévy process with a Lévy measure that depends on two parameters: a positive function $\alpha\left(\cdot\right)$ (called concentration function[3]) over some space $\boldsymbol{\Omega}$ and a base measure H_0 also defined on the space $\boldsymbol{\Omega}$ [6]. Given that the base measure H_0 is continuous, the Lévy measure of the process is:

$$\nu\left(d\theta, d\pi\right) = \alpha\left(\theta\right)\pi^{-1}\exp\left(-\alpha\left(\theta\right)\pi\right)d\pi H_0\left(d\theta\right),$$

on the space $\boldsymbol{\Omega} \times [0, \infty)$, which with respect to θ is a degenerate gamma distribution [6], hence the name gamma process. Draws from the gamma process are discrete and can be represented as:

$$H = \sum_k \pi_k \delta_{\theta_k}, \tag{2.3}$$

where the points $\left(\theta_k, \pi_k\right) \in \boldsymbol{\Omega} \times [0, \infty)$ are draws from the process, an alternative way to write this is $H \sim \Gamma\mathrm{P}\left(\alpha, H_0\right)$. Let $S \in \boldsymbol{\Omega}$, then the total mass $H\left(S\right) = \sum_j \pi_j \mathbb{I}\left(\theta_j \in S\right)$; where $\mathbb{I}\left(\cdot\right)$ is an indicator function that equals to one if the argument inside is true, otherwise it is zero.

Drawing samples from the Dirichlet process $D \sim \mathrm{DP}\left(\alpha, H_0\right)$ [15] is equivalent to normalising the samples from the gamma process, such that:

$$\begin{aligned} H|\alpha, H_0 &\sim \Gamma\mathrm{P}\left(\alpha, H_0\right), \\ D = \frac{H}{H\left(\boldsymbol{\Omega}\right)} &\sim \mathrm{DP}\left(\alpha, H_0\right), \end{aligned}$$

where it should be noted that the Dirichlet process is independent of the normalising constant, i.e., $D \perp H\left(\boldsymbol{\Omega}\right)$

[3] If $\alpha\left(\cdot\right)$ is a constant then this is called a concentration parameter

Dirichlet process mixture model

The DP is used in Bayesian nonparametric for an infinite Bayesian mixture model. Traditionally a Bayesian mixture model with K number of mixtures assumes the following generative model:

$$\begin{aligned}
\boldsymbol{\pi}|\alpha &\sim \text{Dirichlet}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right), \\
\{\theta_k\}_{k=1}^{K} &\overset{i.i.d.}{\sim} H, \\
c_n|\boldsymbol{\pi} &\sim \text{Categorical}\left(\boldsymbol{\pi}\right), \\
y_n|c_n, \theta_{1,\ldots,K} &\sim \mathcal{F}_{\theta_{c_n}},
\end{aligned}$$

where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K] \in [0,1]^K$ be a K-dimensional vector; with $\sum_{k=1}^{K} \pi_k = 1$ and, $\alpha > 0$ is a hyperparameter, $\{\theta_k\}_{k=1}^{K}$ are K parameters independently and identically (i.i.d.) sampled from some distribution H, $c_n \in \{1, \ldots, K\}$ is a variable which indicates which mixtures observation $n \in \{1, \ldots, N\}$ is sampled from and y_n is some observation sampled from some distribution $F_{\theta_{c_n}}$; which is parameterised by θ_{c_n}. This section will explore how the marginal distribution of $c_{1,\ldots,N}$ will behave as the number of clusters K approach to infinite (i.e., $K \to \infty$). The marginal distribution of $c_{1,\ldots,N}$ for finite K is:

$$\begin{aligned}
\mathrm{P}\left(c_{1,\ldots,N}|\alpha\right) &= \int \left(\prod_{n=1}^{N} \mathrm{P}\left(c_n|\boldsymbol{\pi}\right)\right) \mathrm{P}\left(\boldsymbol{\pi}|\alpha\right) d\boldsymbol{\pi} \\
&= \frac{\prod_{k=1}^{K} \Gamma\left(m_k + \frac{\alpha}{K}\right)}{\Gamma\left(\frac{\alpha}{K}\right)^K} \times \frac{\Gamma\left(\alpha\right)}{\Gamma\left(N + \alpha\right)},
\end{aligned}$$

where $m_k = \sum_{n=1}^{N} \mathbb{I}\left(c_n = k\right)$.

The marginal probability of $c_{1,\ldots,N}$ does not depend on the order of the features [8] i.e., classes are *exchangeable* [17]. This results in having multiple $c_{1,\ldots,N}$'s which encode the same class assignments for several objects; an example of this can

$$\begin{aligned}
\mathrm{P}\left(\left[c_{1,\ldots,N}\right] | \alpha\right) &= \sum_{c_{1,\ldots,N} \in \left[c_{1,\ldots,N}\right]} \mathrm{P}\left(c_{1,\ldots,N}\right) \\
&= \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{i=1}^{m_k - 1} \left(i + \frac{\alpha}{K}\right)\right) \frac{\Gamma\left(\alpha\right)}{\Gamma\left(N + \alpha\right)},
\end{aligned}$$

and as $K \to \infty$, the above becomes:

$$\lim_{K \to \infty} \left(\mathrm{P}\left(\left[c_{1,\ldots,N}\right] | \alpha\right)\right) = \alpha^{K_+} \left(\prod_{k=1}^{K^+} \left(m_k - 1\right)!\right) \frac{\Gamma\left(\alpha\right)}{\Gamma\left(N + \alpha\right)}.$$

Construction

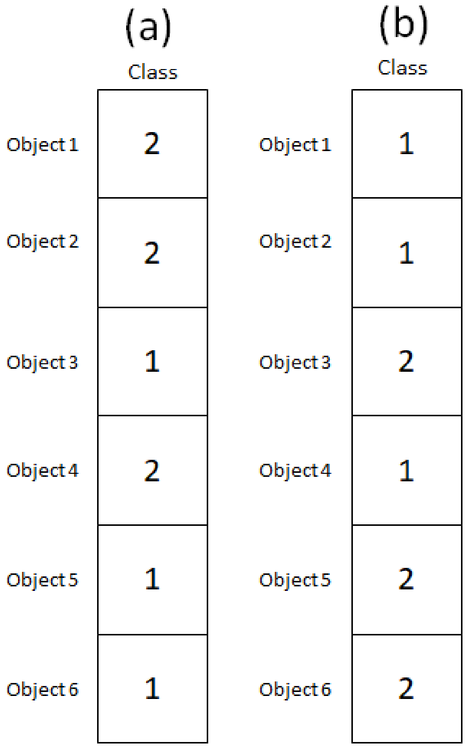
Chinese restaurant process: The *Chinese Restaurant Process* (CRP) [15, 18] is a marginal representation of the infinite Dirichlet-categorical [19]. This process is often explained using a cuisine metaphor, hence the name. Let there exist a Chinese restaurant with an infinite number of tables. The first customer enters the restaurant and takes a seat at the first table. Then the second customer enters, they take a seat at the first table with probability $\frac{1}{\alpha+1}$, and the second table with probability $\frac{\alpha}{\alpha+1}$. In general, the nth customer will take a seat on table k with probability $\frac{m_k}{\alpha+n}$; where m_k is the number of customers at table k, or take a seat at a new (unoccupied by the other $(n-1)$th customers) with probability $\frac{\alpha}{\alpha+n}$.

2.2.2 Beta-Bernoulli Process

Definition

Beta process: The *beta process* [16] is a positive Lévy process with a Lévy measure ν that depends on two parameters: a positive function $\alpha\left(\cdot\right)$ (called concentration function[4]) over $\mathbf{\Omega}$, and a fixed base measure H_0 defined on the space $\mathbf{\Omega}$ [6]. Given that the

[4] If $\alpha\left(\cdot\right)$ is a constant then this is called a concentration parameter



	(a) Class		(b) Class
Object 1	2	Object 1	1
Object 2	2	Object 2	1
Object 3	1	Object 3	2
Object 4	2	Object 4	1
Object 5	1	Object 5	2
Object 6	1	Object 6	2

Figure 2.1: A plot of different class allocation. Both (a) and (b) are different but they both encode the same information; objects 1, 2 & 4 are grouped together and objects 3, 5 & 6 are grouped together.

base measure H_0 is continuous, the Lévy measure of the process is:

$$\nu\left(d\theta, d\pi\right) = \alpha\left(\theta\right)\pi^{-1}\left(1-\pi\right)^{\alpha(\theta)-1} d\pi H_0\left(d\theta\right),$$

on the space $\mathbf{\Omega} \times [0,1]$, which with respect to θ is a degenerate beta distribution [6]; hence the name beta process. Draws from the beta process are discrete and can be represented as:

$$H = \sum_k \pi_k \delta_{\theta_k}, \tag{2.4}$$

where the points $(\theta_k, \pi_k) \in \mathbf{\Omega} \times [0,1]$ are draws from the process, an alternative way to write this is $H \sim \text{BP}\left(\alpha, H_0\right)$. In the context of latent feature models the pair (θ_k, π_k) can be viewed as:

- $\theta_k \in \mathbf{\Omega}$ represents the location for the feature k
- $\pi_k \in [0,1]$ represents the weight associated with feature k, i.e. the probability that an object will 'posses' the kth feature is π_k

See [6, 20] for a more in-depth introduction of the Beta process.

Bernoulli process: The *Bernoulli process* is a positive Lévy process with a Lévy measure:

$$\lambda\left(d\theta, d\pi\right) = \delta_1\left(d\pi\right) H\left(d\theta\right),$$

where H (from equation (2.4)) is a *hazard measure* on the space $\mathbf{\Omega}$ [6]. Assuming the hazzard measure is a collection of infinite points $H = \sum_k \pi_k \delta_{\theta_k}$, then the draws from the Bernoulli process can be represented as:

$$B_n = \sum_k b_{nk} \delta_{\theta_k},$$

where the points $(\theta_k, b_{nk}) \in \mathbf{\Omega} \times \{0,1\}$ are drawn from the process, an alternative way to write this is $B_n \sim \text{BeP}\left(H\right)$. It should be noted that the points b_{nk} are independent Bernoulli variables with π_k representing the probability of $b_{nk} = 1$.

Conjugacy: It is well known that the beta distribution is conjugate to the Bernoulli likelihood, and likewise, it has been shown by [6] that the beta process is conjugate to the Bernoulli process. Let $H \sim \text{BP}(\alpha, H_0)$ and let $B_n \sim \text{BeP}(H)$ for $n \in \{1, \ldots, N\}$ be N independent samples from the Bernoulli process (with hazard measure H), then the posterior update on H is still a beta process:

$$\begin{aligned} H|B_{1,\ldots,N} &\sim \text{BP}(\alpha + N, H_N), \\ \text{where} \quad H_N &= \frac{\alpha}{\alpha + N} H_0 + \sum_j \frac{n_j}{\alpha + N} \delta_{\theta_j}, \end{aligned}$$

where $n_j = \sum_n b_{nj}$ is the number of times in which the atom at location θ_j appears in $B_{1,\ldots,N}$ [6].

In the context of latent feature models, one can interpret $\mathbf{\Omega}$ as being the space of all features, and B_n as an object defined by the features it possess, where the pair (θ_k, b_{nk}) corresponds to B_n possessing feature θ_k if $b_{nk} = 1$ and not possessing feature θ_k if $b_{nk} = 0$. This information can be represented in a binary matrix $\mathbf{Z}$ with N rows to represent the N independent draws from the Bernoulli process and infinite columns to represent the infinite atoms from the beta process:

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ 1 & 0 & 1 & 1 & \cdots \end{bmatrix}. \tag{2.5}$$

Distribution of infinite binary matrices

Finite: Using both the beta and Bernoulli distribution [8] derived the probability distribution over an infinite sparse binary matrix (like the one in equation (2.5)). Let $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$ be a K-dimensional vector; where $\pi_k \in [0, 1]\ \forall k$. The prior distribution over π_k is the Beta distribution:

$$\pi_k|\alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right),$$

where $\alpha > 0$ is a hyperparameter. Let **Z** be a $(N \times K)$ where each element z_{nk} is independently and identically (i.i.d.) sampled from a Bernoulli distribution:

$$z_{nk}|\pi_k \sim \text{Bernoulli}\left(\pi_k\right),$$

where $n \in \{1, \ldots, N\}$, and $k \in \{1, \ldots, K\}$. From a latent feature perspective $z_{nk} = 1$ can be interpreted as object n possessing feature k. The marginal likelihood of **Z** is:

$$\begin{aligned} \mathrm{P}\left(\mathbf{Z}|\alpha\right) &= \prod_{k=1}^{K} \int \left(\prod_{n=1}^{N} \mathrm{P}\left(z_{nk}|\pi_k\right) \right) \mathrm{P}\left(\pi_k|\alpha\right) d\pi_k \\ &= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma\left(m_k + \frac{\alpha}{K}\right)\Gamma\left(N - m_k + 1\right)}{\Gamma\left(N + 1 + \frac{\alpha}{K}\right)}, \end{aligned}$$

where $m_k = \sum_{n=1}^{N} z_{nk}$.

Equivalence Classes: The marginal probability of the binary matrix **Z** does not depend on the order of the features [8] i.e., features are exchangeable. This results in having multiple binary matrices **Z** which encode the same feature assignments for a few objects; an example of this can be seen in Figure 2.2.

To consider the same feature assignments over different feature arrangements, [8] propose to use a left-ordered from function $\mathrm{lof}\left(\cdot\right)$ which maps a binary matrix to its left-ordered form (see Figure 4 in [8]). This function works by ordering the columns of the matrix by the history of the columns, which ensures that the nth row is more significant than the $(n+1)$th row. The history for column k can be calculated using the following:

$$H_k = \sum_{n=1}^{N} 2^{N-n} z_{nk}, \tag{2.6}$$

where H_k denotes the history for column k. Then the cardinality of $[\mathbf{Z}] = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!}$ is the number of matrices that map to the same left-ordered form; where K_h denotes the number of columns with history h.

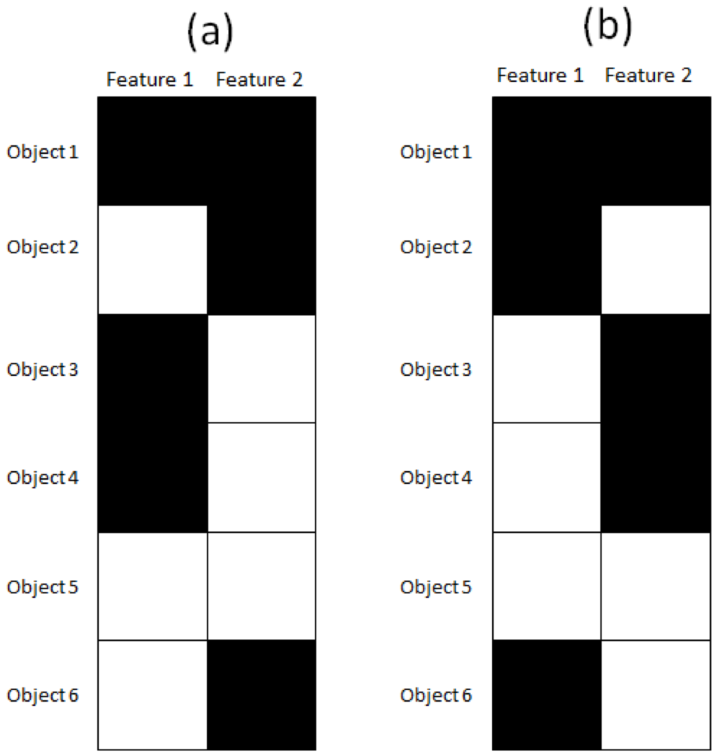


Figure 2.2: A plot of different binary matrix: black cells indicate 1's and white cells indicate 0's. Both (a) and (b) are different matrices but they both encode the same information; objects 1, 3 & 4 are grouped together and objects 1, 2 & 6 are grouped together.

Infinite binary matrix: By using the methods from above, the marginal probability of lof-equivalent class of binary matrices [**Z**] is:

$$\begin{aligned}\mathrm{P}\left([\mathbf{Z}]\,|\alpha\right) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} \mathrm{P}\left(\mathbf{Z}|\alpha\right)\\ &= \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma\left(m_k+\frac{\alpha}{K}\right)\Gamma\left(N-m_k+1\right)}{\Gamma\left(N+1+\frac{\alpha}{K}\right)}.\end{aligned} \tag{2.7}$$

The number of columns K on the matrix **Z** consists of two parts: the first part is the K_0 number of columns with zeros i.e., $m_k = 0$, and the second part is the K_0 number of columns with non-zeros i.e., $m_k > 0$. Therefore, the marginal in equation (2.7) can be re-written as:

$$\begin{aligned}\mathrm{P}\left([\mathbf{Z}]\,|\alpha\right) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} \mathrm{P}\left(\mathbf{Z}|\alpha\right)\\ &= \left[\frac{K!}{\prod_{h=0}^{2^N-1} K_h!}\right] \times \left[\left(\frac{N!}{\prod_{j=1}^{N}\left(j+\frac{\alpha}{K}\right)}\right)^{K}\right] \times \ldots\\ &\quad \ldots \times \left[\left(\frac{\alpha}{K}\right)^{K_+}\right] \times \left[\prod_{k=1}^{K_+} \frac{(N-m_k)!\prod_{j=1}^{N}\left(j+\frac{\alpha}{K}\right)}{N!}\right],\end{aligned}$$

and as $K \to \infty$, the above becomes:

$$\begin{aligned}&\lim_{K\to\infty}\left(\mathrm{P}\left([\mathbf{Z}]\,|\alpha\right)\right) =\\ &\quad\left[\frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!}\right] \times \left[\exp\left(-\alpha H_N\right)\right] \times \left[\prod_{k=1}^{K_+} \frac{(N-m_k)!\,(m_k-1)!}{N!}\right],\end{aligned} \tag{2.8}$$

where $H_N = \sum_{j=1}^{N}\frac{1}{j}$; also known as the Nth harmonic number, and K^+ is the number of columns of **Z** where $m_k > 0$. An alternative view of the equation (2.8) is the probability of observing a matrix that stores draws from the beta-Bernoulli process (like the one in equation (2.5)).

Construction

The Indian Buffet Process (IBP): The *Indian Buffet Process* (IBP) [8] is a marginal representation of the beta-Bernoulli process [6]. This process is often explained using a cuisine metaphor of an Indian buffet with an infinite number of dishes, hence the name. The first customer enters the buffet and takes a serving from each dish till they stop at the Poisson (α) numbered dish. Then the second customer enters, they move along the buffet selecting the kth dish with probability $\frac{m_k}{2}$; where m_k are the number of customers who have tried dish k. After reaching the end of all previously chosen dishes the second customer tries a Poisson $\left(\frac{\alpha}{2}\right)$ number of new dishes. In general, the nth customer will go along the buffet selecting dishes, the probability of selecting the kth dish would $\frac{m_k}{n}$, and then at the end, they try Poisson $\left(\frac{\alpha}{n}\right)$ number of new dishes.

The results of the IBP can be represented using a binary matrix $\mathbf{Z}$ which has N rows (after N number of customers have entered the buffet) an infinite number of columns (as the buffet has an infinite number of dishes). If the nth customer tried the nth dish then $z_{nk} = 1$. The probability of a single lof-equivalent class of binary matrices $[\mathbf{Z}]$ generated from the IBP is:

$$\begin{aligned} \mathrm{P}\left([\mathbf{Z}]\,|\alpha\right) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} \mathrm{P}\left(\mathbf{Z}|\alpha\right) \\ &= \left[\frac{\alpha^{K_+}}{\prod_{h=0}^{2^N-1} K_h!}\right] \times \left[\exp\left(-\alpha H_N\right)\right] \times \left[\prod_{k=1}^{K_+} \frac{(N-m_k)!\,(m_k-1)!}{N!}\right]. \end{aligned} \tag{2.9}$$

Both equation (2.8) and equation (2.9) are the same, this implies that there is an underlying relationship between the IBP and the beta-Bernoulli process; the scope of this report is not to examine this exact relationship, see [6] for more details on this relationship.

Chapter 3

Generalised mixture model to cluster count data

Count data appears in many different domains such as (but not limited to) insurance [21], crime [22], single cell RNA-seq [23–27], document modelling [28, 29]. This is often modelled using distributions from the exponential family of distributions such as the *binomial* distribution, the *Poisson* distribution, and the *negative-binomial* distribution; where the Poisson distribution is a limiting case of the other two. The simple inference of these distributions and computational efficiency comes at the price of strong assumptions about the skewness and dispersion of the probability distributions. Dispersion can be characterised into three categories [30], *underdispersed* when the mean is greater than the variance, *equidispersed* when the mean and variance are equal, and *overdispersed* when the mean is less than the variance. Each category can be modelled using a different distribution, for example, the binomial distribution assumes the data is underdispersed, the Poisson distribution assumes the data is equidispersed, and the negative-binomial distribution assumes the data are overdispersed.

Mixture models on count data provide an important probabilistic framework for decomposing complex distributions into a mixture of simpler components, where each mixture component can be used to represent a distinct cluster [13]. This is why mixtures of binomial, Poisson, and negative-binomial distributions are used in many different applications [31–34]. However, each component will still assume on the dispersion of the data; this is particularly harmful as one cannot statistically assess which distribution is best for which component. For example, if the data are overdispersed then the mixtures of Poisson distributions would not correctly model the data, i.e., the estimated density would be different from the observed density, and a model selection criterion like the Bayesian information criteria (BIC) [14] would encourage more number of clusters than what is actually present in the data.

It is common to alleviate the difficulty of dealing with discrete random variables by using amortised continuous approximations [35]. However, this can somewhat limit the interpretability of individual components or branches of your graphical model. Furthermore, sampling from a continuous approximation of a discrete random variable can have identifiability issues.

3.1 Discrete distributions

This section will highlight the different distributions one can use to model count data.

Binomial distribution

The binomial distribution is a probability distribution which models the number of 'successes' in m number of independent experiments; where each experiment can either be a success (with probability $p \in [0, 1]$) or a 'failure'[1] (with probability $1 - p$) such that:

$$\begin{aligned} y|m, p \sim& \text{binomial}\left(m, p\right), \\ \mathrm{P}\left(y|m, p\right) =& \binom{m}{y} \left(\rho\right)^{y} \left(1 - \rho\right)^{m-y}, \end{aligned} \tag{3.1}$$

where $\binom{m}{y} = \frac{m!}{y!(m-y)!}$ and $y \in \{0, \ldots, m\}$. The mean and variance of which are:

$$\text{mean}\left(y\right) = m\rho,$$

$$\text{variance}\left(y\right) = mp\left(1 - p\right).$$

[1] Often referred to as not a success.

The Bernoulli(p) distribution is a special case of the binomial(m, p) distribution when $m = 1$.

Poisson distribution

The Poisson distribution is a probability distribution that models the counts of an event occurring over a fixed time, for example, the number of people entering a building each hour such that:

$$\begin{aligned} y|\lambda \sim & \text{Poisson}\left(\lambda\right), \\ \text{P}\left(y|\lambda\right) = & \frac{\lambda^y}{y!} \exp\left(-\lambda\right), \end{aligned}$$

where $y \in \{0, 1, 2, \dots\}$. The mean and variance of which are:

$$\begin{aligned} \text{mean}\left(y\right) &= \lambda, \\ \text{variance}\left(y\right) &= \lambda. \end{aligned}$$

Negative-binomial distribution

The negative-binomial distribution is a probability distribution that models the number of 'successes' after r number of 'failures'[2]; this is similar to the binomial distribution, except that instead of fixing the number of experiments, the number of failures is fixed. Like the binomial distribution, the probability of success is $p \in [0, 1]$ and the probability of a failure is $1 - p$ such that:

$$\begin{aligned} y|r, m \sim & \text{negative} - \text{binomial}\left(r, p\right), \\ \text{P}\left(y|r, p\right) = & \binom{r + y - 1}{y} \left(1 - p\right)^r \left(p\right)^y, \end{aligned} \tag{3.2}$$

[2] Or none successes.

where $y \in \{0, 1, 2, \dots\}$. The mean and variance of which are:

$$\begin{aligned} \text{mean}\left(y\right) &= \frac{r\rho}{1-p}, \\ \text{variance}\left(y\right) &= \frac{rp}{\left(1-p\right)^2}. \end{aligned}$$

The Geometric(p) distribution is a special case of the negative-binomial(r, p) distribution when $r = 1$.

Poisson-shifted generalised inverse Gaussian

The Poisson distribution can be extended into many other distributions such as the *Sichel* distribution [36], *Poisson-shifted inverse Gaussian* distribution [37], the *Delaporte* distribution [38] and the *Poisson-shifted exponential* [39]; all of these can be generalised by the *Poisson-shifted generalised inverse Gaussian* (PSGIG) distribution [39]. The PSGIG is derived by first assuming that the data y is sampled from a Poisson distribution parameterised by $\lambda\gamma$ such that:

$$\begin{aligned} y|\lambda, \gamma \sim &\text{Poisson}\left(\lambda\gamma\right), \\ \text{P}\left(y|\lambda, \gamma\right) = &\frac{\left(\lambda\gamma\right)^y}{y!}\exp\left(-\lambda\gamma\right), \end{aligned}$$

where $\lambda > 0$, and $\gamma > 0$ is defined as some "random effect" [39]. A generalised inverse Gaussian (SGIG (α, β, τ)) [36] prior is placed on the random effect variable γ:

$$\text{P}\left(\gamma|\alpha, \beta, \tau\right) = \frac{h^\beta\left(\gamma-\tau\right)^{\beta-1}}{2K_\beta\left(\frac{1}{\alpha}\right)}\exp\left[-\frac{1}{2\alpha}\left(h\left(\gamma-\tau\right)+\frac{1}{d\left(\gamma-\tau\right)}\right)\right], \quad \text{for } \gamma > \tau,$$

where $\alpha > 0$, $\beta \in \mathbb{R}$, $\tau \in [0, 1)$, and:

$$\begin{aligned}
h =& \frac{c}{(1-\tau)}, \\
c =& R_\beta\left(\frac{1}{\alpha}\right), \\
R_\beta(x) =& \frac{K_{\beta+1}(x)}{K_\beta(x)}, \\
K_\beta(x) =& \frac{1}{2}\int_0^\infty t^{\beta-1}\exp\left(-\frac{1}{2}x\left(t+t^{-1}\right)\right)\mathrm{d}t.
\end{aligned}$$

The marginal (with respect to γ) of which results in the PSGIG distribution [39]:

$$\begin{aligned}
\mathrm{P}(y|\lambda,\alpha,\beta,\tau) =& \int \mathrm{P}(y|\lambda,\gamma)\,\mathrm{P}(\gamma|\alpha,\beta,\tau)\,\mathrm{d}\gamma \\
=& \frac{\exp(-\lambda\tau)}{K_\beta\left(\frac{1}{\alpha}\right)} \times \sum_{j=0}^{y}\binom{y}{i}\frac{\lambda^y\tau^{y-i}K_{\beta+i}(\delta)}{y!h^i(\delta\alpha)^{\beta+i}},
\end{aligned}$$

where:

$$\delta = \left(\alpha^{-2} + \frac{2\lambda}{h\alpha}\right)^{1/2},$$

which is a generalisation of:

- The Sichel distribution [36] if $\tau = 0$ [39]
- The Poisson-shifted inverse Gaussian distribution [37] if $\beta = -\frac{1}{2}$ and $\tau = 0$ [39]
- The Delaporte distribution [38] if $\beta > 0$ as $\alpha \to \infty$ which is a generalisation of the Poisson-shifted exponential [39]

Conway–Maxwell–Poisson distribution

The Conway–Maxwell–Poisson (COM-Poisson) distribution [30] is a two-parameter generalisation of the Poisson distribution which does not restrict the dispersion of the data, the PMF of the COM-Poisson(λ, ν) distribution takes the following form:

$$\mathrm{P}(y|\lambda,\nu) = \frac{\lambda^y}{(y!)^\nu}\frac{1}{Z(\lambda,\nu)}, \tag{3.3}$$

where:

$$Z(\lambda, \nu) = \sum_{j=0}^{\infty} \frac{\lambda^j}{(j!)^{\nu}},$$

where $\lambda > 0$ and $\nu \geq 0$. When compared to the Poisson distribution, the additional ν parameter of the COM-Poisson allows for modelling underdispersed data (when $\nu > 1$), equidispersed data ($\nu = 1$), and overdispersed data ($\nu < 1$). The normalising constant $Z(\lambda, \nu)$ of the COM-Poisson cannot be evaluated when $\nu \neq 1$; this requires additional approximations when evaluating the PMF. [40] describes an asymptotic formula to approximate the normalising constant; this only works when $\lambda > 10^{\nu}$, other numerical approximations of the normalising constant have some relative error associated with them too; these approximations must also be used when evaluating the sufficient statistics of the COM-Poisson distribution. Likewise, approximations must be used to learn the parameters of the COM-Poisson; [41] details various methods. One benefit of the COM-Poisson distribution is that it's a generalisation of other well-known distributions:

- If $\nu = 1$ and $\lambda < 1$ then the COM-Poisson(λ, ν) is equivalent to the Geometric$(1-\lambda)$ distribution.
- If $\nu = 1$ then the COM-Poisson(λ, ν) is equivalent to the Poisson(λ) distribution.
- If $\nu \to \infty$ then the COM-Poisson(λ, ν) is equivalent to the Bernoulli$(\frac{\lambda}{\lambda+1})$ distribution.

3.2 Panjer distribution

Panjers' recursion

Let $S = \sum_{n=1}^{N} X_n$ be a sum of $N \in \mathbb{N}_0$ independent random variables $X_n \in \mathbb{N}_0$; where $n \in \{1, \ldots, N\}$, and N is independent to $\{X_n\}_{n=1}^{N}$. The PMF of the compounded random variable $S \in \mathbb{N}_0$ is [42]:

$$\mathrm{P}\left(S=s\right)=\begin{cases} g_0 & if \quad s=0 \\ \sum_{n=1}^{\infty} g_n f^{n*}\left(s\right) & if \quad s>0 \end{cases}, \tag{3.4}$$

where $g_i = \mathrm{P}\left(N=i\right)$, and $f^{n*}\left(s\right)$ is the nth convolution of some density $f\left(\cdot\right)$ which is used to evaluates the probability of X_n. Panjer [43] proposed a recursive formula to overcome the computationally expensive convolution in equation (3.4) if random variable N is sampled from the Panjer family of distributions:

$$\mathrm{P}\left(s|a,b\right)=\left(a+\frac{b}{y}\right)\mathrm{P}\left(s-1\right), \tag{3.5}$$

where $s>0$, $\{a,b\}\in\mathbb{R}$ such that $a+b\geq 0$ and $\mathrm{P}\left(0\right)=1-\sum_{s=1}^{\infty}\mathrm{P}\left(s\right)$. The Panjer family (denoted as Panjer$(a,b,0)$) of distributions consists of distributions like the binomial distribution, the Poisson distribution, the negative-binomial distribution, and many more. The main purpose of the Panjer family of distributions is to check whether the Panjer recursion can be applied or not, however, the striking connection between various probability distributions is ignored, which motivates us to entirely focus on the Panjer family of distributions.

Definition

The Panjer recursion can be generalised to Panjer(a,b,n), where the recursion from equation (3.5) still applies except $\mathrm{P}\left(k\right)=0 \quad \forall k<n$; see [44]. The Panjer$(a,b,0)$ recursion probabilities and parameters can be written in a all-in-one formula called the Panjer distribution [44], the Panjer distribution has the following PMF:

$$\mathrm{P}\left(y|\lambda,\eta\right)=\left(1+\frac{\lambda}{\eta}\right)^{-\eta}\frac{\lambda^y}{y!}\prod_{i=0}^{y-1}\frac{\eta+i}{\eta+\lambda}, \tag{3.6}$$

where $y \in \mathbb{N}_0$, $\lambda > 0$ and $\eta \in \{[-\infty, -\lambda) \cup (0, \infty]\}$ are distributional parameters. The mean and variance of which are:

$$\begin{aligned} \text{mean}\left[y\right] &= \lambda, \\ \text{variance}\left(y\right) &= \lambda\left(1 + \frac{\lambda}{\eta}\right). \end{aligned} \tag{3.7}$$

It can be argued that the exponential family consists of distributions like the binomial distribution, the Poisson distribution, and the negative-binomial distribution and therefore there is no need for the Panjer family (except for checking if Panjers' recursion is applicable) and by extension the Panjer distribution. However, it is not that the binomial distribution belongs to the exponential family, but it is the binomial distribution with a known 'number of trials' (the m parameter from equation (3.1)) that belongs to the exponential family. Likewise, it is not that the negative-binomial distribution belongs to the exponential family, but it is the negative-binomial distribution with a known 'number of failures until the experiment is stopped' (the r parameter from equation (3.2)) that belongs to the exponential family. These subtle changes in the definitions of these distributions show that the special cases of these distributions are unified within the exponential family of distributions; these special cases allow us to utilise conjugacy for more efficient posterior updates on the parameters. However, the exact distributions are only unified within the Panjer family using the Panjer distribution; this is explored in Appendix A.1. Therefore, unsurprisingly the Panjer distribution does not belong to the exponential family (see Appendix A.2), but its special cases are; i.e., binomial distribution with a known 'number of trials'. The result of this is that we cannot use conjugacy to update the posterior distribution of the parameters of the Panjer distribution. However, the benefit of not having to confine the Panjer distribution into the exponential family form is that the parameters of the distribution are flexible (i.e., not restricted to be within a certain domain; see Appendix A.1) which alleviates any assumption on the dispersion of the data.

Advantages of modeling using Panjer distribution

Probability distributions like the binomial distribution, the negative-binomial distribution, and Poisson distribution make a strong a-priori assumption on the dispersion of the data; this is due to the coupling of the parameters when the mean and variance of these distributions are evaluated. These restrictions could hinder the ability to correctly model some data if the 'true' dispersion of the data is unknown. On the other hand, the Panjer distribution does not restrict modelling of the data according to the dispersion of the data i.e., the mean is independent of the variance.

The COM-Poisson distribution can also be used to model in a situation where the 'true' dispersion of the data is unknown as it can generalise distributions like the geometric distribution (for overdispersed data) and Bernoulli distribution (for underdispersed data). However, the Panjer distribution can also generalise distributions like the geometric distribution (as it's a special case of the negative-binomial distribution) and the Bernoulli distribution (as it's a special case of the binomial distribution). However, approximations with relative error must be used to evaluate the PMF or the sufficient statistics of the COM-Poisson distribution; this is because of the normalising constant of the COM-Poisson (see equation (3.3)). On the other hand, the Panjer distribution does not need any approximations when evaluating the PMF or sufficient statistics.

An experiment was conducted to demonstrate how well the probability distributions above can estimate the density of differed types (by dispersion) of data sets, three data sets of size 1000 were generated: the first data set was underdispersed (generated using the binomial distribution), the second data set was equidispersed (generated using the Poisson distribution) and the third data set was overdispersed (generated using the negative-binomial distribution). The absolute difference between the estimated density and the empirical density can be seen in Table 3.1, and a visualisation of the estimated PMF and the histogram of the data can also be seen in Figure 3.1. Table 3.1 shows that the Panjer distribution models binomial data the best, both the Panjer and Poisson distribution model Poisson data the best, and both the Panjer and negative-binomial distributions model the negative-binomial data the best; Figure 3.1 also reflect the same

Table 3.1: The absolute error between the estimated density and empirical density of four count distribution on modelling three different (by dispersion) data sets.

	Data		
Distribution	Binomial	Poisson	Negative-binomial
Binomial	0.103	0.432	1.148
Poisson	0.330	**0.077**	1.013
Negative-binomial	0.519	0.216	**0.250**
Panjer	**0.064**	**0.077**	**0.250**

results. From this, we can conclude that irrespective of the dispersion of the data, the Panjer distribution does better than the binomial distribution, Poisson distribution, and negative-Binomial distribution.

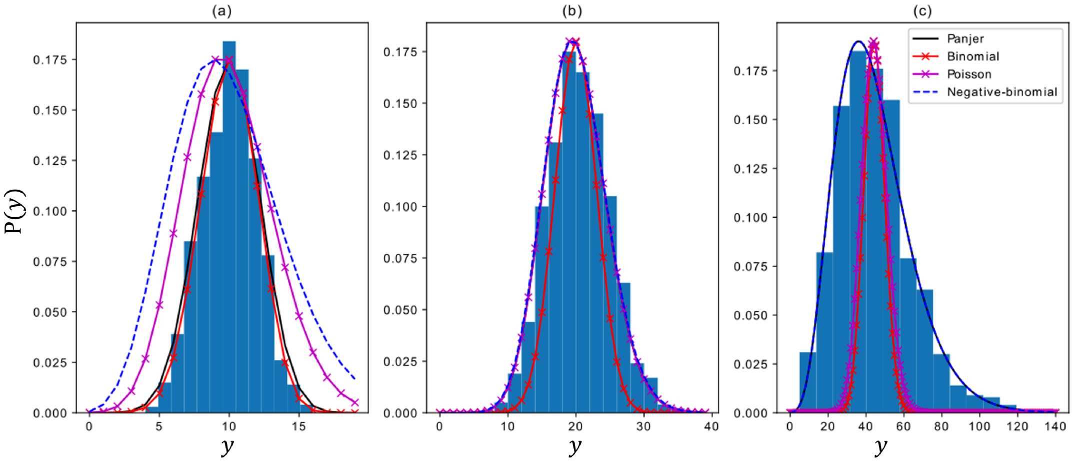


Figure 3.1: Plot of the PMF of four distributions over three different data sets; (a) data sampled from the binomial distribution, (b) data sampled from the Poisson distribution and (c) data sampled from the negative-binomial distribution.

3.2.1 Parameter updates

Maximum likelihood is a scheme used to learn the parameters of a probability distribution given some data; this is done by maximising the likelihood (or the log-likelihood) of the observed data with respect to the parameters of the distribution. Let $\{y_n\}_{n=1}^{N}$ be a set of N i.i.d. samples from the Panjer distribution parameterised by λ and η (see equation (3.6)). The data likelihood is:

$$\mathrm{P}\left(\{y_n\}_{n=1}^N|\lambda,\eta\right) \propto \prod_{n=1}^{N}\left[\left(1+\frac{\lambda}{\eta}\right)^{-\eta}\frac{\lambda^{y_n}}{y_n!}\prod_{i=0}^{y_n-1}\frac{\eta+i}{\eta+\lambda}\right],$$

and the data log-likelihood is:

$$\mathcal{L} \propto \sum_{n=1}^{N}\left[-\eta\ln\left(1+\frac{\lambda}{\eta}\right)+y_n\ln\left(\lambda\right)-\ln\left(y_n!\right)+\sum_{i=0}^{y_n-1}\left(\ln\left(\frac{\eta+i}{\eta+\lambda}\right)\right)\right].$$

The maximum likelihood update for λ is $\lambda_{ML} = \frac{1}{N}\sum_{n=1}^{N} y_n$; this is obtained by differentiating the log-likelihood with respect to λ and setting it to zero (see Appendix A.3 for the derivations). However, the maximum likelihood update for η is not available in closed-form, this is because there is no closed-form solution of the derivative of the log-likelihood with respect to η when set to zero; see Appendix A.3 for the derivations. Therefore, the maximum likelihood solution for η solves the equation $f\left(\eta_{ML}\right) = 0$ where:

$$f\left(\eta\right) = -N\left(\ln\left(\mathrm{sgn}\left(\eta\right)\left(\lambda+\eta\right)\right)-\ln\left(\mathrm{sgn}\left(\eta\right)\eta\right)\right)+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right],$$

where $\mathrm{sgn}\left(\cdot\right)$ is a sign function, such that:

$$\mathrm{sgn}\left(x\right) = \begin{cases} -1 & \text{if} \quad x < 1 \\ 0 & \text{if} \quad x = 0 \\ 1 & \text{if} \quad x > 1 \end{cases} \cdot$$

Empirical results suggest that iterative method like the Newton-Raphson method (initialised using method of moments (MOM) update) can learn η_{ML} in a few iterations; the MOM update is:

$$\eta_{MOM} = \frac{\mathrm{mean}\left(\{y_n\}_{n=1}^N\right)^2}{\mathrm{variance}\left(\{y_n\}_{n=1}^N\right)-\mathrm{mean}\left(\{y_n\}_{n=1}^N\right)},$$

where mean $\left(\{y_n\}_{n=1}^N\right)$ is the sample mean, and variance $\left(\{y_n\}_{n=1}^N\right)$ is the sample variance.

3.3 Panjer mixture model

Let $\{y_n\}_{n=1}^N$ be N one-dimensional[3] discrete observations sampled from a mixture of K Panjer distributions, such that the probability of some observation y_n given all parameters is:

$$\mathrm{P}\left(y_n|\boldsymbol{\theta}\right) = \sum_k \pi_k \mathrm{P}\left(y_n|\lambda_k,\eta_k\right), \tag{3.8}$$

where $\boldsymbol{\theta} = \{\lambda_1,\dots,\lambda_K,\eta_1,\dots,\eta_K\}$ contains all the parameters of the K Panjer distributions and $\{\pi_k\}_{k=1}^K$ are the mixing weights; such that $\sum_{k=1}^K \pi_k = 1$.

3.3.1 Inference

Two schemes are proposed to learn the parameters of the Panjer mixture model, both schemes require K-dimensional latent variables $\{\mathbf{z}_n\}_{n=1}^N$. Let the latent variable (or cluster assignment) $\mathbf{z}_n$ be a random variable that indicates which mixture an observation was sampled from; such that if $z_{nk} = 1$ then observation n was sampled from cluster k. Additional constraints such as $\sum_{i=1}^K z_{ni} = 1$ are imposed on the latent variables and $\mathrm{P}\left(z_{nk}=1\right) = \pi_k$; where π_k are the mixing weights from equation (3.8). Using the latent variables, the probability of some observation y_n given all parameters is:

$$\mathrm{P}\left(y_n|\mathbf{z}_n,\boldsymbol{\theta}\right) = \prod_k \mathrm{P}\left(y_n|\lambda_k,\eta_k\right)^{z_{nk}}.$$

The first scheme to infer all the parameters is the expectation-maximisation (EM) algorithm [45], this scheme iteratively switches between the E-step and the M-step:

- **E-step**: Take expectation of the latent variables $\{\mathbf{z}_n\}_{n=1}^N$ with respect to the posterior distribution.

[3] This is done for notational convenience, see Appendix A.4 for a D-dimensional generalisation.

- **M-step**: Maximise the complete data log-likelihood with respect to the other parameters.

This is repeated till convergence; the full algorithm is described in Appendix A.4.1.

An alternative to the EM algorithm is the maximisation-maximisation (MM) algorithm (this is often called hard EM [45]), this scheme iteratively switches between the M-step and the M-step:

- **M-step**: Maximise the complete data log-likelihood with respect to the latent variables $\{\mathbf{z}_n\}_{n=1}^N$.
- **M-step**: Maximise the complete data log-likelihood with respect to the other parameters.

This is repeated till convergence; the full algorithm is described in Appendix A.4.2.

Both schemes still impose $\sum_{i=1}^K z_{ni} = 1$, but they differ on the domain of each z_{nk}; the EM assumes $z_{nk} \in [0,1]$, whereas the MM assumes $z_{nk} \in \{0,1\}$. To understand this subtle difference the performance of both schemes was compared on synthetic data of varying size and parameters; it was observed that the complete data log-likelihood obtained by MM was always larger than the complete data log-likelihood obtained by the EM. Furthermore, the MM had a lower variation of information (VI) [46] score than the EM; this score estimates the information shared between the true clustering and the estimated clustering where a zero is a complete agreement; this is similar to the Binder's loss [47]; see [48] for more information. This shows that the MM scheme is better than the EM scheme for this problem, the reason for this may be that the EM assigns 'soft clustering' while MM assigns 'hard clustering'.

3.4 Experiments

Three data sets will be used to assess how well the Panjer mixture model (PanjerMM) is able to learn the density of some observations. For comparison, the binomial mixture (BMM), the Poisson mixture model (PMM), and the negative-binomial mixture model

(NBMM) will also be applied to the same data sets. The MM scheme (described in section 3.3.1) will be used to learn the parameters of each mixture model, each scheme will run till the difference in the complete data log-likelihood is less than 10^{-4} and restarted ten times with random initialisation; results from the best run will be reported.

Synthetic

The first data set consists of 3000 synthetic observations generated from three equal-sized component mixture models; the first component is a negative-binomial distribution, the second component is a Poisson distribution and the third component is a binomial; the data generating process can be seen in Algorithm 1, and the data can be seen in Figure 3.2.

Algorithm 1 Generating synthetic data

```
for $n \leftarrow 1$ to $3000$
    Sample: $k \sim$ multinomial $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$
    if $k = 1$
        Sample: $y_n$ $\sim$negative-binomial$(5, 0.1)$
    else if $k = 2$
        Sample: $y_n$ $\sim$Poisson$(20)$
    else
        Sample: $y_n$ $\sim$binomial$(20, 0.5)$
Output: Observations: $y_{1,\ldots,N}$
```

In Table 3.2 we report the density estimation error; which is the absolute error between the estimated density and empirical density, the percentage accuracy of the clustering; which describes the percentage of points that were correctly assigned to the correct mixture component, and the variation of information (VI) [46]; which calculates the information shared between the true clustering and the estimated clustering where a zero is a complete agreement.

From Table 3.2 we can see that the performance of both the NBMM and PMM was poor, Figure 3.2 shows that they both failed to capture the middle and right-most component correctly; this is probably due to their assumptions on dispersion. Surprisingly, the BMM managed to do quite well but it failed to correctly estimate the tails of the component densities, hence the slightly higher density estimation error. The PanjerMM did

Table 3.2: Performance of different count data mixture models. We report the density estimation error which is the absolute error between the estimated density and observed density, the accuracy of each mixture model, and the variation of information (VI) of each model.

	Result		
Mixture model	Density estimation error	Accuracy	Variation of information
Binomial	0.268	97.4%	0.334
Poisson	0.346	95.2%	0.473
Negative-binomial	0.425	92.7%	0.622
Panjer	**0.103**	**97.7%**	**0.305**

the best in terms of learning the location and spread of the data for each component; note that the slight discrepancy in the accuracy occurs because the tails of each component overlap, hence it is difficult to cluster data that lies on the tails of each component. The dispersion restrictions imposed by BMM, PMM, and NBMM hinder the mixture models from correctly modelling the data, whereas the flexibility of the PanjMM enables modelling in situations where each component has different dispersion.

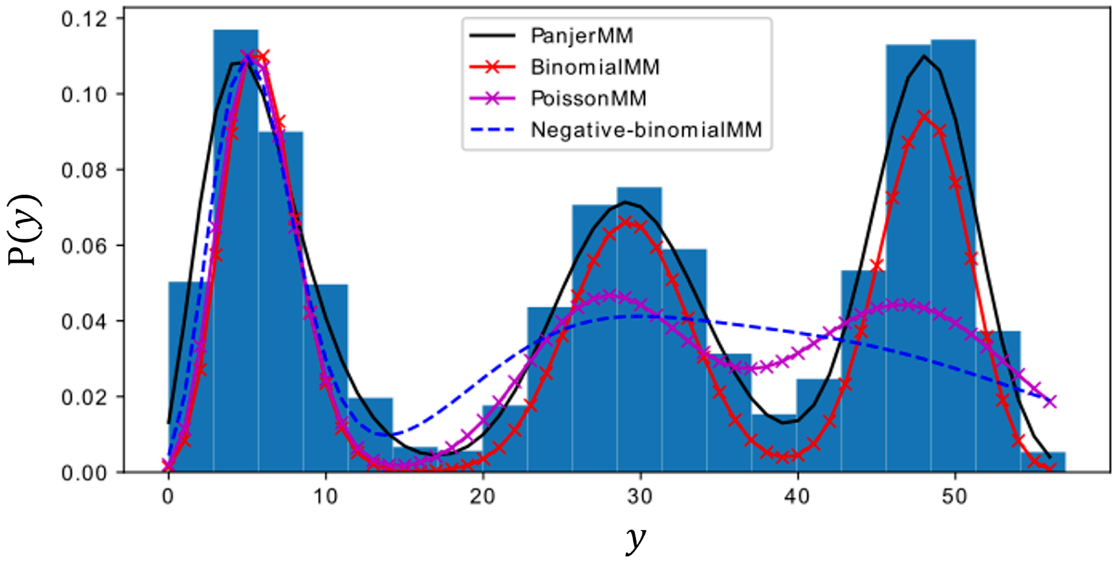


Figure 3.2: A comparison of the PMF learnt by four different mixture models on data generated using Algorithm 1.The left-most component is sampled from the negative-binomial distribution, the middle component is sampled from the Poisson distribution and the right-most component is sampled form the binomial distribution.

Table 3.3: The performance of different count data mixture models on mortality data from UK Office for National statistics [49]. We report the density estimation error which is the absolute error between the estimated density and observed density, the accuracy of each mixture model, and the variation of information (VI) of each model.

	Result		
Mixture model	Density estimation error	Accuracy	Variation of information
Binomial	0.564	99%	0.14
Poisson	0.433	**100%**	**0.00**
Negative-binomial	0.465	99%	0.14
Panjer	**0.407**	**100%**	**0.00**

Mortality data

The second data set is from the UK Office for National Statistics, the data consist of the weekly deaths in England and Wales between two age groups [49] in 2010; each age group represents a component in a two-component mixture model. The first component is the number of deaths of children under the age of one; this has a mean of 33.4 and a variance of 27.7 (underdispersed), the second component is the number of deaths of children between the ages of one to fourteen; this has a mean of 11.7 and a variance of 14.2 (overdispersed); the data can be seen in Figure 3.3. The goal is to not only cluster the data correctly but also learn the correct density of the data.

Results from Table 3.3 show that in terms of clustering, both the BMM and NBMM have some misclassifications; this can be seen in Figure 3.3 where BMM underestimated the density 'in-between' the two components, and the NBMM overestimated the density 'in-between' the two components. Both the PMM and PanjerMM do equally well in terms of clustering the data, however, the PMM slightly underestimates the density in the right component, hence it has a higher density estimation error.

Weaving data

The third data set is a count of the number of breaks in a fixed length of yarn (also called a loom) [50]; a histogram plot of the data can be seen in Figure 3.4. The experiment was run with three different numbers of components (K) to estimate the density of the data,

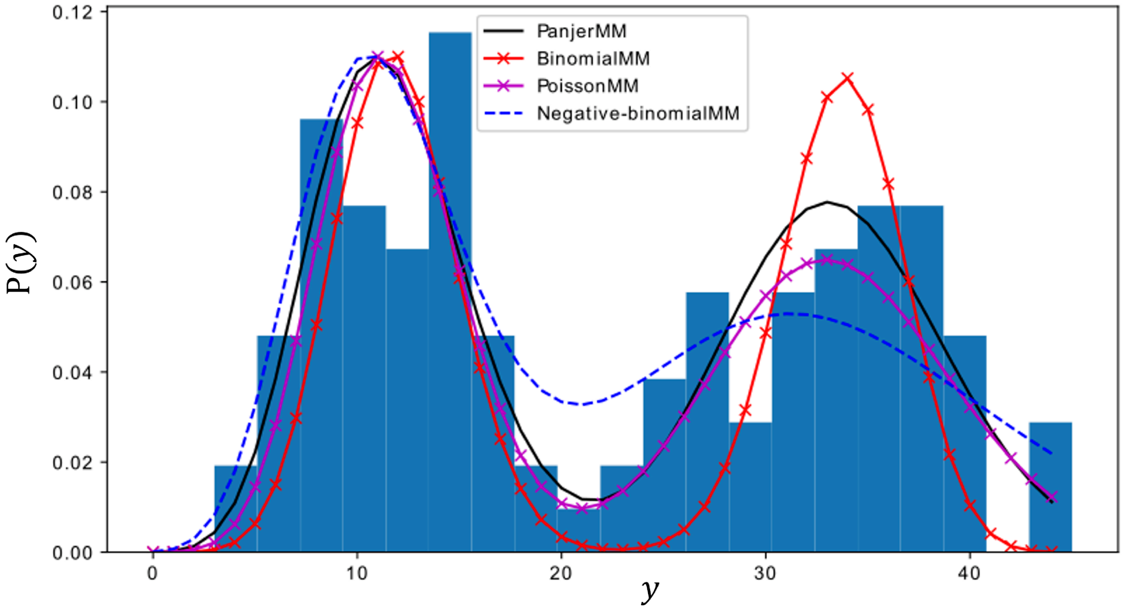


Figure 3.3: A comparison of the PMF learned by four different mixture models on mortality data from England and Wales between two age groups in 2010. The left component represents the number of deaths in children between the ages of one to fourteen and the right component represents the number of deaths for children under the age of one.

the results of these can be seen in Table 3.4, and the plot of the densities with $K = 2$ number of components can be seen in Figure 3.4. Results from Table 3.4 suggest that the PanjerMM has the lowest density estimation error irrespective of the number of components, except with $K = 2$ where NBMM is equally as good; this is because the 'true' data generating process may have been the two-component NBMM (i.e., both components are overdispersed). From this, we can see that the flexibility of the PanjerMM allows modelling even if each component has the same type of dispersion.

Table 3.4: The performance of different count data mixture models on waving data [50]. We report the density estimation error which is the absolute error between the estimated density and observed density over different number of components K.

	Density estimation error		
Mixture model	$K = 1$	$K = 2$	$K = 3$
Binomial	1.133	0.874	0.802
Poisson	0.991	0.786	0.733
Negative-binomial	0.871	**0.730**	0.743
Panjer	**0.755**	**0.730**	**0.727**

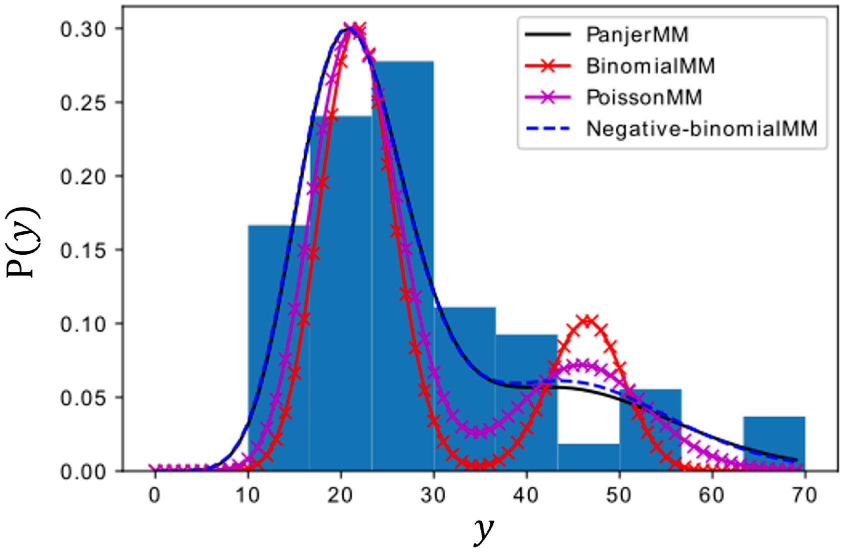


Figure 3.4: A comparison of the PMF learned by four different two-component mixture models on the number of breaks in yarn during weaving.

3.5 Discussion

In this chapter, we have studied both clustering and density estimation of count data using mixture models. The assumptions on the dispersion of data often restrict the most common distributions like the Poisson; this is due to the coupling of the distributional parameters. This becomes more problematic in mixture models, as each component of the mixture is assumed to have the same dispersion, this is clear in all experiments. The Panjer distribution does not make any assumptions about the dispersion of the data, this motivates its use in a mixture model; we call this the Panjer mixture model. In terms of clustering and density estimation, the Panjer mixture model is far superior to the other mixture models. We also proposed two different schemes to infer the parameters of the mixture model; where we saw that the discrete nature of the data space preferred the MM scheme over the traditional EM scheme.

Currently, the D-dimensional multivariate Panjer distribution is assumed to be a product of D univariate Panjer distributions; although this overcomes the problems of other D univariate distributions it fails to account for dependencies among the dimensions. One promising direction of the Panjer distribution (and its mixture model) is to have a multivariate extension of the distribution which is not a product of D univariate distributions; this can be achieved as the parameters of the distribution are not coupled.

Chapter 4

Robustifying mixture models using the maximum mean discrepancy

It is well known that one popular application for mixture models is clustering data using a probabilistic framework; where each mixture component can be used to represent a distinct cluster [13] or a building block for clusters at different levels of abstraction [51]. Mixture distributions also provide a formal framework for specifying assumptions about: the cluster shape [47, 52, 53]; the partition topology [54, 55]; the expected partition process [56] and others. Inference in mixtures is generally intractable so we resort to iterative optimisation algorithms to achieve maximum likelihood estimation [57–60], finding a point estimate of the posterior mode [61, 62], or simulate the desired posterior using a wide variety of *Markov Chain Monte Carlo* (MCMC) techniques [63–65]. However, this assumes that the model is correctly specified (or not *misspecified*). If this is not the case, and the mixture model is misspecified then cluster assignments will be incorrect. Therefore, in this chapter, we propose a method to learn cluster assignments using a mixture model given that there is misspecification.

4.1 Problem definition

Let $\{\mathbf{y}_n\}_{n=1}^N \in \mathcal{Y}$ be a set of N D-dimensional observed data points in some data space $\mathcal{Y}$, where each data point $\mathbf{y}_n$ is sampled from some unknown distribution $\mathcal{F}^*$. In modeling one assumes that $\mathcal{F}^* \overset{\mathrm{d}}{=} \mathcal{F}_{\boldsymbol{\theta}}$ i.e., each true data generating distribution can be represented using some parametric density $\mathcal{F}_{\boldsymbol{\theta}}$ such as an exponential family, or variational autoencoder (VAE) with parameter(s) $\boldsymbol{\theta} \in \Theta$; where if $\mathcal{F}_{\boldsymbol{\theta}}$ generates multivariate Gaussian data then $\boldsymbol{\theta}$ can represent the mean, the covariance matrix or both. In the Bayesian paradigm these parameters have the following posterior distribution:

$$\mathrm{P}\left(\boldsymbol{\theta} | \{\mathrm{y}_n\}_{n=1}^{N}\right) \propto \underbrace{\mathrm{P}\left(\boldsymbol{\theta}\right)}_{prior} \times \underbrace{\mathrm{P}_{\boldsymbol{\theta}}\left(\{\mathrm{y}_n\}_{n=1}^{N} | \boldsymbol{\theta}\right)}_{likelihood}, \tag{4.1}$$

where $\mathrm{P}_{\boldsymbol{\theta}}\left(\cdot\right)$ is the probability function[1] of the data generating distribution $\mathcal{F}_{\boldsymbol{\theta}}$. The parameter(s) $\boldsymbol{\theta}$ can be inferred in a number of different ways: (1) Gibbs sampling where the inferred parameter is sampled from the posterior distribution conditioned on other parameter(s) using an MCMC, (2) Maximum likelihood of the observed data where the inferred parameter(s) maximise the likelihood portion of equation (4.1), and (3) Maximum-a-posteriori where the inferred parameter(s) maximise the posterior distribution.

However, the assumption $\mathcal{F}^* \overset{\mathrm{d}}{=} \mathcal{F}_{\tilde{\boldsymbol{\theta}}}$ may not always be true as the true data generating distribution is unknown; where $\tilde{\boldsymbol{\theta}}$ denotes the inferred parameter(s). This will result in inferring incorrect distributional parameter(s) $\tilde{\boldsymbol{\theta}}$ which as a result will estimate the incorrect density. For example, a Gaussian distribution will not correctly model skew-Gaussian data, this becomes more problematic if the data consists of outliers; see Figure 4.1. We call this problem misspecified likelihood.

4.2 Proposed maximum mean discrepancy pseudo-point marginal

The main problem with misspecified likelihood is that there is always some uncertainty if the assumed data generating distribution is equivalent to the true data generating distribution (i.e., $\mathcal{F}^* \overset{\mathrm{d}}{=} \mathcal{F}_{\tilde{\boldsymbol{\theta}}}$). Instead of representing the chosen likelihood using a distribution parameterised by some parameter(s) $\boldsymbol{\theta}$ (i.e., $\mathrm{P}_{\boldsymbol{\theta}}\left(\cdot\right)$), we propose to use 'pseudo-

[1] This can be a probability mass function if the data space is discrete, or a probability density function if the data space is continuous.

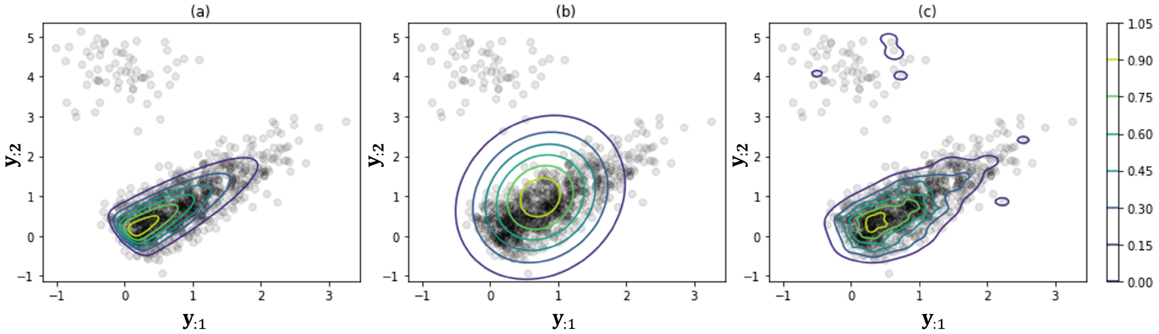


Figure 4.1: Plot of the estimated densities on skew-Gaussian data with outliers; (a) Observed data and the true data skew-Gaussian density, (b) Density estimated by a multivariate Gaussian distribution, and (c) Density estimated by the proposed maximum mean discrepancy pseudo-point marginal.

points' to represent the likelihood. These D dimensional pseudo-points $\{\mathbf{u}_m\}_{m=1}^{M} \in \mathcal{Y}$ have the following empirical density:

$$\tilde{\mathrm{P}}_{\mathbf{u}} = \frac{1}{M}\sum_{m=1}^{M}\delta_{\mathbf{u}_m},$$

where $\delta_{\mathbf{u}}$ is a Dirac measure at $\mathbf{u} \in \mathcal{Y}$. The M pseudo-points $\{\mathbf{u}_m\}_{m=1}^{M}$ are treated as model parameters and are optimised to minimise some statistical distance $d(\cdot,\cdot)$ between $\tilde{\mathrm{P}}_{\mathbf{u}}$ and the true probability function P^* (the probability function of the true data generating distribution $\mathcal{F}^*$), such that:

$$\tilde{\mathbf{u}}_m = \underset{\mathbf{u}_m}{\operatorname{argmin}}\left[d\left(\mathrm{P}^*, \tilde{\mathrm{P}}_{\mathbf{u}}\right)\right].$$

In other words $\tilde{\mathrm{P}}_{\mathbf{u}}$ aims to mimic P^*. However, P^* is unknown but we can approximated this using the N observed data points:

$$\mathrm{P}^* \approx \tilde{\mathrm{P}}_{\mathbf{y}} = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{y}_n}. \tag{4.2}$$

4.2.1 Choosing a statistical distance $d(\cdot,\cdot)$

There are many different families of statistical distances/divergences that can be used to compute $d(\cdot,\cdot)$, for example the *Ali-Silvey* [66] distance which is a generalisation of distances/divergence like the Kullback-Liebler (KL) divergence, Hellinger distance etc. However, it has been observed in [67] that computing the Ali-Silvey distance in high dimensions can be tough. An alternative way to define $d\left(\cdot,\cdot\right)$ is to compare the summary statistics (i.e., mean and variance) between the two densities, this can be done using integral probability metrics (IPM) [68], which is a generalisation of distances such as the Wasserstein distance (also called the earth mover distance), Dudley metric, total variation, maximum mean discrepancy (MMD), etc [67]. It has been observed that the MMD is a robust distance to model misspecification [69, 70], furthermore, the MMD also allows one to decide what summary statistic(s) should be used to compare densities. Therefore, the statistical distance $d\left(\cdot,\cdot\right)$ will be defined using the MMD, which is defined as [71]:

$$\begin{aligned} d\left(\tilde{\mathrm{P}}_{\mathbf{y}},\tilde{\mathrm{P}}_{\mathbf{u}}\right) = \mathrm{MMD}\left[\tilde{\mathrm{P}}_{\mathbf{y}},\tilde{\mathrm{P}}_{\mathbf{u}}\right] &= \sup_{f\in\mathcal{G}}\left\{\int f\left(Y\right)\tilde{\mathrm{P}}_{\mathbf{y}}\left(dY\right) - \int f\left(Y\right)\tilde{\mathrm{P}}_{\mathbf{u}}\left(dY\right)\right\} \\ &= \sup_{f\in\mathcal{G}}\left\{\left\langle f, \mu_{\tilde{\mathrm{P}}_{\mathrm{y}}} - \mu_{\tilde{\mathrm{P}}_{\mathrm{u}}}\right\rangle_{\mathcal{H}}\right\} \\ &= \left\|\mu_{\tilde{\mathrm{P}}_{\mathrm{y}}} - \mu_{\tilde{\mathrm{P}}_{\mathrm{u}}}\right\|_{\mathcal{H}} \end{aligned} \tag{4.3}$$

where Y is a random variable on some topological space $\mathcal{Y}$, $\mathcal{G}$ are a class of function used to compute summary statistics in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, and $\mu_{\tilde{\mathrm{P}}_{\mathrm{y}}}$ and $\mu_{\tilde{\mathrm{P}}_{\mathrm{u}}}$ are the 'mean embedding' of the distributions $\tilde{\mathrm{P}}_{\mathbf{y}}$ and $\tilde{\mathrm{P}}_{\mathbf{u}}$ respectively in the RKHS. An added advantage of the MMD is that equation (4.3) can be approximated using samples from the two densities:

$$\begin{aligned} \mathrm{MMD}^2\left[\tilde{\mathrm{P}}_{\mathbf{y}},\tilde{\mathrm{P}}_{\mathbf{u}}\right] = &\frac{1}{N^2}\sum_{i,j}k\left(\mathbf{y}_i,\mathbf{y}_j\right) + \frac{1}{M^2}\sum_{i,j}k\left(\mathbf{u}_i,\mathbf{u}_j\right) \\ &- \frac{2}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}k\left(\mathbf{u}_m,\mathbf{y}_n\right), \end{aligned} \tag{4.4}$$

where here $k\left(\cdot,\cdot\right)\in\mathcal{H}$ is a kernel in the reproducing kernel Hilbert space $\mathcal{H}$. The MMD can be used as a probability metric if the kernel is characteristic [67]; for example, the Gaussian kernel.

Therefore, the proposed model uses M pseudo-points $\{\mathbf{u}_m\}_{m=1}^{M}$ to approximate the true density of the data by using the MMD; hence the name maximum mean discrepancy pseudo-point marginal. In the proposed MMD pseudo-point marginal, the probability likelihood of an observation $\mathbf{y}_n$ is:

$$\mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m\}_{m=1}^{M}\right) = \lambda \exp\left(-\lambda \times \mathrm{MMD}^2\left[\delta_{\mathbf{y}_n}, \tilde{\mathrm{P}}_{\mathbf{u}}\right]\right), \tag{4.5}$$

where $\delta_{\mathbf{y}_n}$ is a Dirac measure at $\mathbf{y}_n \in \mathcal{Y}$, and $\lambda > 0$ is a distributional parameter (typically set to one), and:

$$\mathrm{MMD}^2\left[\delta_{\mathbf{y}_n}, \tilde{\mathrm{P}}_{\mathbf{u}}\right] = k\left(\mathbf{y}_n, \mathbf{y}_n\right) + \frac{1}{M^2}\sum_{i,j} k\left(\mathbf{u}_i, \mathbf{u}_j\right) - \frac{2}{M}\sum_{m=1}^{M} k\left(\mathbf{u}_m, \mathbf{y}_n\right).$$

The advantage of the proposed MMD pseudo-point marginal can be seen in Figure 4.1, where the MMD pseudo-point marginal (with $M = 10$ and Gaussian kernel) is not affected by any outliers, moreover, no assumptions on the true data generating distribution are assumed.

Related work

One closely related solution to this type of problem is to replace the *likelihood* term of Bayes‘ rule with a *loss function*; this is called generalised Bayesian inference (GBI) which has a *generalised posterior* [72, 73]:

$$\mathrm{P}\left(\boldsymbol{\theta} | \{\mathbf{y}_n\}_{n=1}^{N}\right) \propto \underbrace{\mathrm{P}\left(\boldsymbol{\theta}\right)}_{prior} \times \underbrace{\exp\left(-\omega \ell\left(\{\mathbf{y}_n\}_{n=1}^{N}, \boldsymbol{\theta}\right)\right)}_{loss\ function}, \tag{4.6}$$

where the prior distribution encodes the prior belief of $\boldsymbol{\theta}$ in the absence of any data, $\{\mathbf{y}_n\}_{n=1}^{N}$ is a set of N data points, $\omega \geq 0$ is the learning rate, and $\ell\left(\{\mathbf{y}_n\}_{n=1}^{N}, \boldsymbol{\theta}\right)$ is a loss function. If $\omega = 1$ and $\ell\left(\{\mathbf{y}_n\}_{n=1}^{N}, \boldsymbol{\theta}\right)$ is the negative log-likelihood then equation (4.6) is equivalent to the traditional Bayesian posterior; see [74] for a more in-depth introduction.

The rationale behind this generalised posterior is that as data increases, summarising robustly the posterior is further masked by sensitivity to misspecification, i.e., the prior distribution becomes dominated by the likelihood as data increase. This is further exacerbated if the data has some misspecification. One solution to the data likelihood dominating the posterior is to temper the likelihood [75]; where the loss function in equation (4.6) $\ell\left(\{\mathbf{y}_n\}_{n=1}^{N}, \boldsymbol{\theta}\right)$ is the negative log-likelihood and $\omega \geq 0$ controls the prominence of the prior, where:

- If $\omega = 0$, then the posterior is equivalent to the prior
- If $0 < \omega < 1$, then the prior is given more precedence than it would when compared to the standard Bayesian posterior
- If $\omega = 1$, then the posterior is equivalent to the standard Bayesian posterior
- If $\omega > 1$, then the likelihood is given more precedence than it would when compared to the standard Bayesian posterior

Miller and Dunson [76] showed that likelihood tempering can be taught as a relaxation which averages over all the support within the KL divergence neighbourhood of the unknown population distribution [76]. Furthermore, the loss function of the generalised posterior can be replaced with any discrepancy metric; this will allow one to evaluate the likelihood of data given the parameter $\boldsymbol{\theta}$ in a robust manner [77]. However, all of these approaches differ from the proposed MMD pseudo-point marginal in three ways: (1) They place significance on inferring the parameter $\boldsymbol{\theta}$ in the presence of misspecification over learning the correct data generating distribution, (2) They assume that the prior distribution contains the 'true' distribution of $\boldsymbol{\theta}$ which is then used to find the 'true' value, and (3) All parameters of the model are coupled for example in a Gaussian model, the mean and variance will be updated in the same fashion.

4.3 Mixtures of maximum mean discrepancy pseudo-point marginal

Let $\{\mathbf{y}_n\}_{n=1}^N$ be a set of N D-dimensional observed data points which are multi-modal and generated by a mixture of K components:

$$\begin{aligned}\boldsymbol{\pi}|\alpha &\sim \text{Dirichlet}\left(\frac{\alpha}{K},\dots,\frac{\alpha}{K}\right),\\ c_n|\boldsymbol{\pi} &\sim \text{Categorical}\left(\boldsymbol{\pi}\right),\\ \mathbf{y}_n|c_n &\sim \mathcal{F}_{c_n}^*,\end{aligned}$$

where K denotes the number of mixture components which can be fixed, random or infinite; $\pi = [\pi_1, \dots, \pi_K]$ denote some mixing parameters typically assuming $\sum_{k=1}^K \pi_k = 1$; $\{\pi_k\}_{k=1}^K \in [0, 1]$, $\alpha > 0$ is a hyperparameter, $c_n \in \{1, \dots, K\}$ is a variable which indicates which mixtures observation n is sampled from, and $\{\mathcal{F}_k^*\}_{k=1}^K$ are K *true* unknown data generating distribution; or mixture components. Each mixture component can be interpreted as a district cluster which results in $c_n \in \{1, \dots, K\}$ representing which 'cluster' observation n belongs to. We assume that $\mathcal{F}_k^* \overset{\text{d}}{=} \mathcal{F}_{\boldsymbol{\theta}_k}$ for $k \in \{1, \dots, K\}$, i.e., each true data generating mixture component $\mathcal{F}_k^*$ can be represented using a parametric density $\mathcal{F}_{\boldsymbol{\theta}_k}$ which parameterised by some parameter(s) $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}$;. This results in the following data-generating process:

$$\begin{aligned}\boldsymbol{\pi}|\alpha &\sim \text{Dirichlet}\left(\frac{\alpha}{K},\dots,\frac{\alpha}{K}\right),\\ \{\boldsymbol{\theta}_k\}_{k=1}^K &\overset{i.i.d.}{\sim} H,\\ c_n|\boldsymbol{\pi} &\sim \text{Categorical}\left(\boldsymbol{\pi}\right),\\ \mathbf{y}_n|\theta_{c_n} &\sim \mathcal{F}_{\boldsymbol{\theta}_{c_n}},\end{aligned}$$

where the distribution parameters $\{\boldsymbol{\theta}_k\}_{k=1}^K$ are K i.i.d. samples from some distribution H. The parameters $\{\boldsymbol{\pi}, c_1 \dots, c_N, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}$ can be inferred in a variety of different ways, for example: (1) The expectation-maximisation (EM) scheme which first takes the expectation (E-step) of the latent variables $\{c_n\}_{n=1}^N$ with respect the posterior distribution and then maximises (M-step) the complete-data likelihood with respect to the remaining parameters, (2) Gibbs sampling which iteratively samples each parameter conditioned on the other parameters using an MCMC, and (3) Maximum-a-posterior where each inferred value of a parameter maximises the posterior distribution of the parameter.

However, as discussed in Section 4.1, the assumption $\mathcal{F}_k^* \overset{\mathrm{d}}{=} \mathcal{F}_{\tilde{\boldsymbol{\theta}}_k}$ may not always be true $\forall\, k$ as the true data generating distributions $\{\mathcal{F}_k^*\}_{k=1}^K$ are unknown; where $\{\tilde{\boldsymbol{\theta}}_k\}_{k=1}^K$ denotes the inferred parameters. Furthermore, this also implies that all components are the same distributions, i.e., mixtures of Gaussian's, which again is not always true. This not only leads to inferring the incorrect distributional parameters $\{\tilde{\boldsymbol{\theta}}_k\}_{k=1}^K$, but the inferred clustering $\{\tilde{c}_k\}_{k=1}^K$ will also be incorrect. We call this misspecified mixture model.

To solve the problem faced by misspecified mixture models, we propose to use the MMD pseudo-points marginals on mixture models, where each mixture component is represented using M pseudo-points, we call this the mixtures of MMD pseudo-point marginal. The posterior probability of c_n under the proposed model is:

$$\mathrm{P}\left(c_n = k | \mathbf{y}_n, \{\mathbf{u}_m^{(k)}\}_{m=1}^M\right) \propto \pi_k \times \lambda \exp\left(-\lambda \times \mathrm{MMD}^2\left[\delta_{\mathbf{y}_n}, \tilde{\mathrm{P}}_{\mathbf{u}^{(k)}}\right]\right).$$

An expectation-maximisation (EM) [45] algorithm to infer all the parameters of mixtures of MMD pseudo-point marginal is described in Appendix B.1.

4.4 Results

The proposed mixtures of MMD psudo-points marginal are applied on three different data sets to see if it is able to learn more accurate cluster assignments when compared to other variants of the Gaussian mixture models.

4.4.1 Synthetic data set

Skew data

The first data set consists of $N = 1000$ two-dimensional synthetic sampled from a two-component mixture model seen in Algorithm 2, where:

$$\boldsymbol{\mu}_1 = [-1.5, 0]\,, \boldsymbol{\mu}_2 = [-2, 0]\,,$$

and $\mathbf{I}_2$ is a (2×2) identity matrix. The data can be seen in Figure 4.2.

Algorithm 2 Generating synthetic data

```
for $n \leftarrow 1$ to $1000$
  Sample: $k \sim$ multinomial $\left(\frac{1}{2}, \frac{1}{2}\right)$
  if $k = 1$
    Sample: $\mathbf{y}_n \sim \mathcal{MVN}(0, \mathbf{I}_2)$
    Set: $\mathbf{y}_n = |\mathbf{y}_n| + \boldsymbol{\mu}_1$
  if $k = 2$
    Sample: $\mathbf{y}_n \sim \mathcal{MVN}(0, \mathbf{I}_2)$
    Set: $\mathbf{y}_n = -|\mathbf{y}_n| - \boldsymbol{\mu}_2$
```

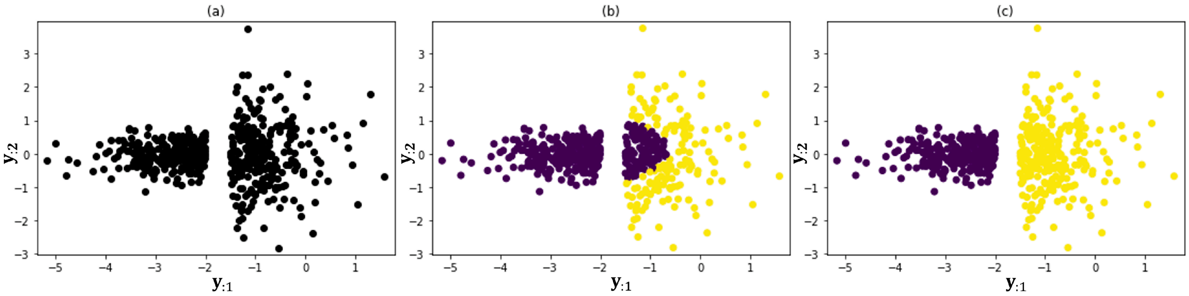


Figure 4.2: Plot of inferred clustering on two dimensional synthetic data; **(a)** Observed data, **(b)** Clustering inferred by Gaussian mixture model, and **(c)** Clustering inferred by mixtures of MMD pseudo-point marginal.

The mixtures of MMD pseudo-point marginal is compared with three other models: the standard GMM, likelihood-tempered Gaussian mixture model (LT-GMM) and the posterior bootstrap Gaussian mixture model (PB-GMM) [78]. Each model was trained on 80% of the data set and tested on the remaining 20%; All mixture models were trained using the EM algorithm and a Gibbs sampler; the EM was run till the difference in the log-likelihood is less than some pre-defined threshold $\epsilon = 1e^{-4}$, and the Gibbs sampler was run for 5,000 iterations and the MAP clustering was selected by maximising the log-likelihood (after burn-in). The mixtures of MMD pseudo-point marginal have 10 pseudo-points per mixture, which uses the Gaussian kernel to compute the MMD. At each iteration the LT-GMM was given a choice between how much information to use from the prior and the data to update the Gaussian means, this is done using two additional pieces of information, first a parameter $\zeta \in [0, 1]$ which controls how much 'trust' there is within the data over the prior; three different runs with $\zeta \in \{0.1, 0.5, 0.9\}$ were executed and results from the best runs are reported (results from all experiments can be found in Appendix B.2), the second requirement was having some 'prior'; in this case the 'true' Gaussian means were given as the prior. One must note that the TL-GMM is very similar to the coarsening posterior [76] except the entire data likelihood is not assumed to be 'corrupted'. At each iteration, the PB-GMM would use the existing parameters to generate 'pseudo-samples' which would then be used to find multiple Gaussian means for each component k (see Algorithm 1 in [78]); this requires the following parameters: number of bootstrap samples $B \in \{100, 1000\}$, number of 'pseudo-samples' $T \in \{100, 10000\}$ and concentration parameter $c \in \{1, 10, 100\}$, multiple runs over different values were executed and results from the best runs are reported (results from all experiments can be found in Appendix B.2).

	Variants of mixture models			
Data	Standard GMM	MMD pseudo-points marginal	LT GMM	PB GMM
Train Data (EM)	81 ± 1	$\mathbf{100 \pm 0}$	85 ± 2	-
Train Data (Gibbs)	80 ± 1	$\mathbf{100 \pm 0}$	84 ± 3	82 ± 1
Test Data (EM)	78 ± 1	$\mathbf{100 \pm 0}$	81 ± 1	-
Test Data (Gibbs)	76 ± 1	$\mathbf{100 \pm 0}$	81 ± 2	80 ± 2

Table 4.1: Performance of different mixture models on synthetic data set. Each model was trained on 80% of the data set and tested on the remaining 20%. The average accuracy and one standard deviation from 20 different experiments are reported.

Results from Table 4.1 suggest that both the Gibbs and EM have similar results over all models. As expected the standard GMM overlaps the two densities of the two components (see Figure 4.2) and therefore does the worst. The PB-GMM does marginally better than the standard Gaussian MM but fails still fails to learn the correct clusters.

As expected, the LT-GMM does reasonably better as a low ζ value allows the model to trust the true prior more when updating the means; however, the Gaussian assumption prevents the model from learning the correct clusters. Unsurprisingly, the mixtures of MMD pseudo-points marginal does the best as the pseudo-points are flexible enough to learn different structures.

Circles data

The second data set is the circles' data set from the SCIKIT LEARN library [79], which consists of two clusters that overlap (see Figure 4.3). We used the GMM to see how well it can cluster this type of data, and unsurprisingly it failed to learn the correct clusters. This is because the two clusters (in terms of their first moment) overlap, whereas the Gaussian MM attempts to separate the first moments (see Figure 4.4). Unsurprisingly, the mixtures of MMD pseudo-points marginal is able to learn the correct clusters (see Figure 4.3), this is because the RKHS gives the model the ability to learn different strictures; this can be seen in Figure 4.4 where the pseudo-points model the structure of each component.

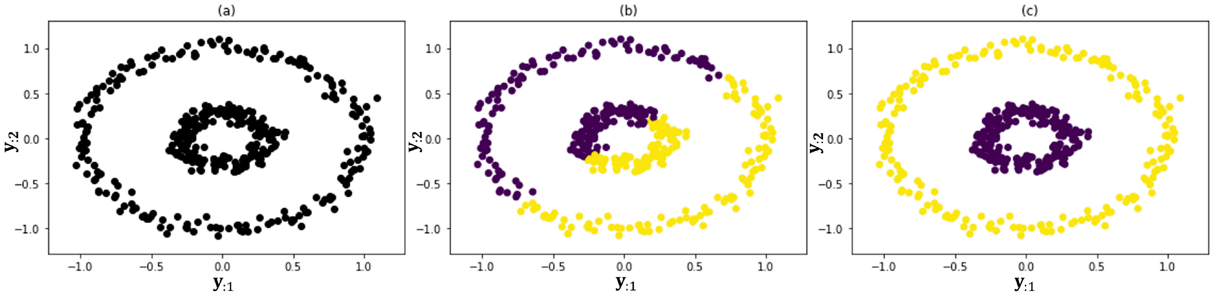


Figure 4.3: Plot of inferred clustering on circles data set; **(a)** Observed data, **(b)** Clustering inferred by Gaussian mixture model with 50% accuracy, and **(c)** Clustering inferred by mixtures of MMD pseudo-point marginal with 100% accuracy.

4.4.2 Single cell ATAC-seq data

The second experiment consists of clustering *assay for transposase-accessible chromatin using sequencing* (ATAC-seq) data from 10X Genomes internal single-cell demonstration data set of *peripheral blood mononuclear cells* (PBMCs) from a healthy donor [80]. We

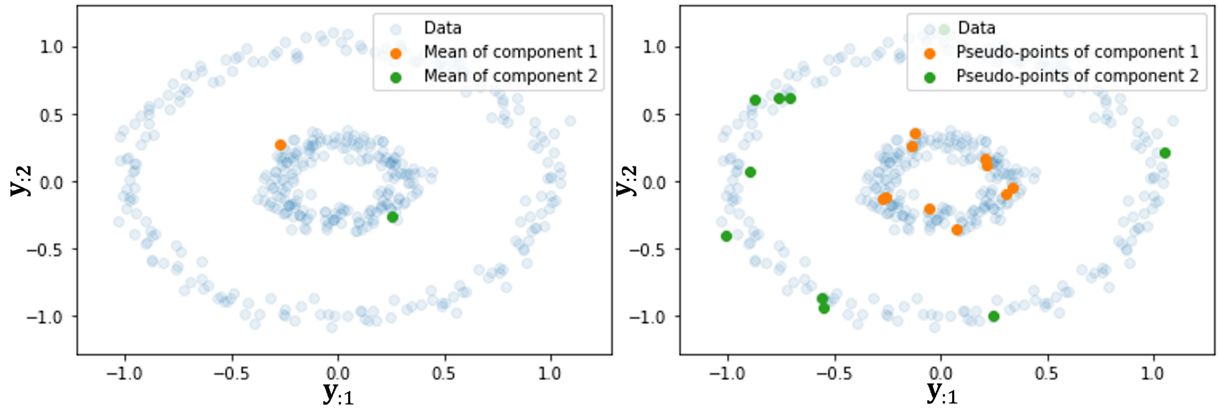


Figure 4.4: Plot of inferred parameters for circles data set; **Left**: The mean of each component inferred by the Gaussian MM, **Right**: The pseudo-points of each component of inferred by mixtures of MMD pseudo-point marginal.

use a negative-binomial mixture model (NBMM) which is a common choice for this data type; for each component, the parameters are r_k- the number of failures until the experiment is stopped and p_{kd}- the success probability of the dimension d of the kth mixture. The single nuclei ATAC data set was pre-processed using Cell Ranger ATAC 1.2.0. Cell Ranger detected 482 nuclei with 17,879 median fragments per cell and a total of 47,843 peaks. Peaks that had less than 1% of the cells active or more than 75% of the cells active were removed, this gave us a dataset with $N = 482$ cells and $D = 26,216$ peaks. Three different dimensionality techniques are used in Figure 4.5 to visualise the results in two-dimensions; the first column uses *principal component analysis* (PCA) on the log-transformed data, the second column uses *t-distributed stochastic neighbourhood* (t-SNE) [2], and the final column uses *uniform manifold approximation* (UMAP) [81]. The first row of Figure 4.5 shows the estimated cell-specific sequencing depth, which is calculated using: $\alpha_n = \frac{1}{D}\sum_d y_{nd}$ and is then re-normalised to $[0, 1]$ where 0 indicates low sequencing depth and 1 indicates high sequencing depth. The main challenge of clustering ATAC-seq data is cell-specific sequencing depth as it introduces inherent bias. To avoid clustering the sequencing depth we propose to use the mixtures of MMD pseudo-point marginal, in which the MMD kernel is the modified cosine similarity:

$$k\left(\mathbf{y}_i, \mathbf{y}_j\right) = \left|\frac{\mathbf{y}_i \cdot \mathbf{y}_j}{\|\mathbf{y}_i\| \, \|\mathbf{y}_j\|}\right| \in [0, 1],$$

where $k(\mathbf{y}_i, \mathbf{y}_j) = 1$ if and only if $\mathbf{y}_i = \mathbf{y}_j$; note that in this situation we'd want to maximise the MMD criterion. We compared the standard NBMM with the mixtures of MMD pseudo-point marginal and K-means clustering algorithm. K is also inferred given the data, this is done by first placing a binomial distribution prior to K, and then evaluating the most likely value for K given the data and model; this is done for standard NBMM and mixtures of MMD pseudo-point marginal, but not for K-means clustering algorithm as it's not probabilistic. The standard NBMM only learns two clusters (see Figure 4.5); these clusters clearly capture the effect of cell-specific sequencing depth, rather than the global structure. Results for the K-means suffer from a similar problem to the standard NBMM. However, the mixtures of MMD pseudo-point marginal identify the three correct clusters, all of which are not fooled by the sequencing depth.

4.5 Discussion

In this chapter, we propose an alternative method to robustify models when the likelihood is misspecified. We propose a novel model which represents the likelihood using pseudo-points that have a density similar to samples drawn from the true data-generating process. Inferring these pseudo-points and computing any probability is done via the maximum mean discrepancy which compares the distance between the family of summary statistics between two distributions using the RKHS; hence the proposed model is called maximum mean discrepancy pseudo-point marginal. The proposed model is used to cluster data in a mixture model setup, we show how other model agnostic approaches such as likelihood tempering fail to cluster data when the model is slightly misspecified. We also demonstrate the utility of the MMD by clustering more complicated data where the moments of the components overlap in the Euclidean space but are in fact well separated in the RKHS. This property of the RKHS was also useful in clustering high-dimensional single-cell ATAC-seq data. One future direction of the proposed model is to apply in different sets ups, currently, this has only been used in mixture models, but it has the potential to be used in other domains such as dimension-

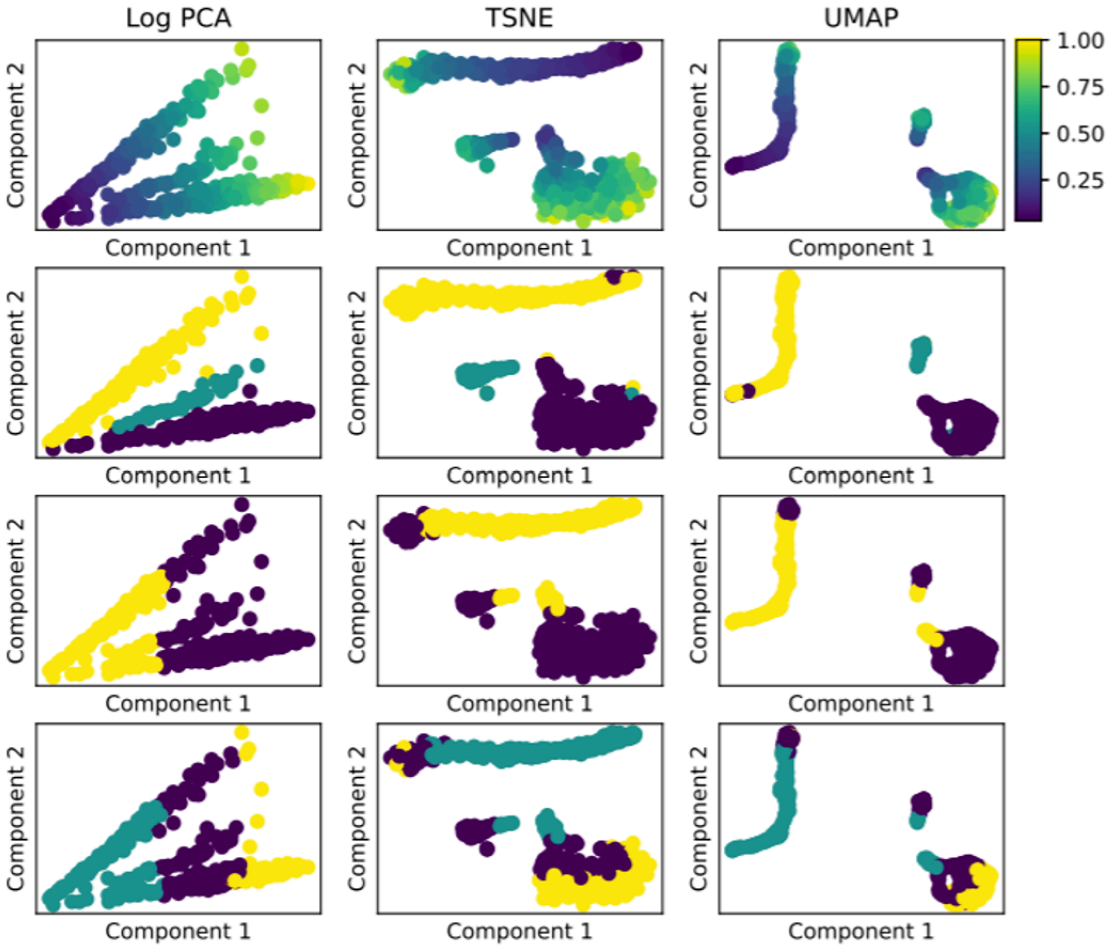


Figure 4.5: Plot of single cell ATAC-seq data used. **Left column**: Log transformed PCA plot of the data. **Middle column**: TSNE plot of the data. **Right column**: UMAP plot of the data. **1st row**: Coloured according to cell specific sequencing depth. **2nd row**: Data coloured according to clustering inferred by mixtures of MMD pseudo-point marginal. **3rd row**: Data coloured according to clustering inferred by standard NBMM. **4th row**: Data coloured according to clustering inferred by K-means.

ality reduction or training variational autoencoders.

Chapter 5

Piecewise linear dimensionality reduction

[1]

Linear dimensionality reduction methods such as *factor analysis* (FA) [82] (defined in Section 5.4.1) and *principal component analysis* (PCA) [83] (defined in Section 5.5) are a mainstay of high-dimensional data analysis, due to their simple geometric interpretation and attractive computational properties. Both FA and PCA can be seen as matrix decomposition techniques that aim to explain the dependence structure among high-dimensional observations through a decomposing of the covariance matrix of the data, which is positive-definite [84]. As data increases in size and complexity, the assumption that linear components are a linear combination of *all* of the original variables becomes increasingly restrictive. Therefore, it is essential to equip these models with the ability to control the number of unique components used to represent each data point. This is critical since it will naturally separate: (1) Components that explain large variance percentages for a small subset of the data, and (2) Spurious components which explain a small variance percentage for a potentially larger subset of the data. This formulation is natural in the context of data visualisation and dimensionality reduction, where natural constraints on the feature representation for each data point occur in visualisation, normally, points are reduced to two or three dimensions, in dimensionality reduction.

[1] Full paper is available here.

5.1 Linear Gaussian model

The linear Gaussian LVM assumes that all observed data $\mathbf{y} \in \mathbb{R}^D$ can be decomposed in the following way:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \ + \boldsymbol{\mu} \ + \ \boldsymbol{\epsilon}, \tag{5.1}$$

where $\mathbf{W} \in \mathbb{R}^{D \times K}$ is a transformation matrix, $\mathbf{x} \in \mathbb{R}^K$ are unknown latent variables, $\boldsymbol{\mu} \in \mathbb{R}^D$ is a mean (offset) vector and $\boldsymbol{\epsilon}$ describes the model noise (typically Gaussian). Various widely-used techniques are obtained by making different assumptions on the prior distributions of $\mathbf{x}$, $\mathbf{W}$ and $\boldsymbol{\epsilon}$; these will be explored in the following sections.

5.2 Factor analysis

Factor analysis (FA) assumes that the observed data is a linear combination of some unobserved factors. The data-generating process uses the linear Gaussian LVM form from equation (5.1) in the following way:

$$\begin{aligned}
\mathbf{w}_k &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}_D\right), \\
\boldsymbol{\epsilon}_n &\sim \mathcal{MVN}\left(\mathbf{0}, \text{diag}\left(\boldsymbol{\sigma}^2\right)\right), \\
\mathbf{x}_n &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_x^2 \mathbf{I}_K\right), \\
\mathbf{y}_n &= \mathbf{W}\mathbf{x}_n + \boldsymbol{\epsilon}_n,
\end{aligned} \tag{5.2}$$

where $\mathbf{w}_k \in \mathbb{R}^D$ for $k \in \{1, \ldots, K\}$ are factor loading vectors, all which make the $(D \times K)$ factor loading matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$, $\mathbf{I}_K$ is an identity matrix of size K, $\mathcal{MVN}(\cdot, \cdot)$ is the multivariate Gaussian distribution, $\{\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_D^2), \sigma_w^2, \sigma_x^2\}$ are parameters which control the variance of each variable, and $n \in \{1, \ldots, N\}$ indexes a single data point. Each observation $\mathbf{y}_n$ is assumed to be a linear combination of all factors, this is quite restrictive as learning the global structure of the dataspace is prioritised over any local structure of the data space.

5.3 Latent feature factor analysis models

In this section, we derive flexible latent feature FA models that leverage weakly constrained feature allocations and allow us to model a wide range of sparse FA models. We demonstrate that by uninformative discrete distributions or constrained distributions without replacement, one can derive simple modelling alternatives which have favourable properties such as the ability to represent both sparse and dense factor loadings. Furthermore, this prevents the FA from prioritising the global structure of the data space over the local structure of the dataspace [85–88]. The addition of latent variable generalises the FA data generating process from equation (5.2) into:

$$\begin{aligned} \mathbf{w}_k|\sigma_w^2 &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}_D\right), \\ \boldsymbol{\epsilon}_n|\boldsymbol{\sigma}^2 &\sim \mathcal{MVN}\left(\mathbf{0}, \text{diag}\left(\boldsymbol{\sigma}^2\right)\right), \\ \mathbf{x}_n|\sigma_x^2 &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_x^2 \mathbf{I}_K\right), \\ \mathbf{z}_n &\sim \mathcal{F}\left(\cdot\right) \\ \mathbf{y}_n &= \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n, \end{aligned} \tag{5.3}$$

where $\mathbf{w}_k \in \mathbb{R}^D$ for $k \in \{1, \dots, K\}$ are factor loading vectors, all which make the $(D \times K)$ factor loading matrix $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_K]$, $\mathbf{I}_K$ is an identity matrix of size K, $\mathcal{MVN}(\cdot, \cdot)$ is the multivariate Gaussian distribution, $\{\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_D^2), \sigma_w^2, \sigma_x^2\}$ are parameters which control the variance of each variable, $n \in \{1, \dots, N\}$ indexes a single data point, $\odot$ denotes the *Hadamard product*, also known as the element-wise or *Schur product* and $\mathbf{z}_n$ is a K dimensional binary latent variable sampled from some distribution $\mathcal{F}(\cdot)$. In the following section, we'll see different assumptions on the distribution $\mathcal{F}(\cdot)$ results in different variants of FA; however, it should be noted that if $\mathbf{z}_n$ is full of ones, then the data generating process in equation (5.3) is equivalent to the data generating process in equation (5.2).

5.3.1 Infinite sparse FA

The *infinite sparse* FA (isFA) [85] model deviates from the data generation process in equation (5.3) in three ways: (1) It assumes an IBP prior on the latent variables $\{\mathbf{z}_n\}_{n=1}^N$ with the concentration parameter $\alpha > 0$, (2) The model noise has a single variance parameter σ^2 (i,e., $\boldsymbol{\epsilon}_n \sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2\mathbf{I}_D\right)$, and (3) The latent variables have fixed variance $\sigma_x^2 = 1$ (i.e., $\mathbf{x}_n \sim \mathcal{MVN}\left(\mathbf{0}, \mathbf{I}_K\right)$).

Inference

By using the data generation process in equation (5.3), the joint likelihood of the isFA is:

$$\begin{aligned} \mathrm{P}\big(\{\mathbf{y}_n\}_{n=1}^N, \{\mathbf{W}_k\}_{k=1}^K, \{\mathbf{x}_n\}_{n=1}^N, \{\mathbf{z}_n\}_{n=1}^N | \boldsymbol{\theta}\big) \\ = \prod_{n=1}^N \left(\mathrm{P}\left(\mathbf{y}_n | \mathbf{W}, \mathbf{x}_n, \mathbf{z}_n, \sigma^2\right) \prod_{k=1}^K \mathrm{P}\left(x_{kn}\right) \mathrm{P}\left(z_{kn}|\alpha\right)\right) \\ \times \prod_{k=1}^K \mathrm{P}\left(\mathbf{w}_k|\sigma_W^2\right), \end{aligned}$$

where $\boldsymbol{\theta} = \{\alpha, \sigma^2, \sigma_W^2\}$ consists of all the remaining parameters. A straightforward Gibbs sampler can be used to infer all the parameters of the isFA model; however, a more scalable variational inference algorithm can also be used on the isFA model [86].

The posterior distribution over the latent variables x_{kn} for which its respective $z_{kn} = 1$ is sampled from a Gaussian:

$$x_{kn}|\mathbf{y}_n, \mathbf{z}_{-kn}, \{\mathbf{w}_k\}_{k=1}^K, \sigma^2 \sim \mathcal{N}\left(\frac{\mathbf{w}_k^T \boldsymbol{\epsilon}_{-kn}}{\sigma^2 + \mathbf{w}_k^T \mathbf{w}}, \frac{\sigma^2}{\sigma^2 + \mathbf{w}_k^T \mathbf{w}}\right),$$

where $\boldsymbol{\epsilon}_{-kn} = \mathbf{y}_n - \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right)$ with $z_{kn} = 0$, or the noise associated with nth point and kth feature.

The posterior distribution over the kth factor loading $\mathbf{w}_k$ is a D-dimensional multivariate Gaussian:

$$\mathbf{w}_k|\mathbf{Y},\mathbf{X},\mathbf{W}^{-k},\sigma_W^2,\sigma^2 \sim \mathcal{MVN}\left(\frac{\sigma_W^2}{\mathbf{x}_k\mathbf{x}_k^T\sigma_W^2+\sigma^2}\mathbf{E}_{-k}\mathbf{x}_k^T,\left(\frac{\mathbf{x}_k\mathbf{x}_k^T}{\sigma^2}+\frac{1}{\sigma_W^2}\right)\mathbf{I}_D\right),$$

where $\mathbf{W}^{-k}$ is the matrix $\mathbf{W}$ without the k column set to zero, $\mathbf{x}_k$ is the kth row of the matrix $\mathbf{X}$, $\mathbf{E}_{-k}$ is $(\mathbf{Y}-\mathbf{W}(\mathbf{X}\odot\mathbf{Z}))$ with $\mathbf{w}_k=\mathbf{0}$, and $\mathbf{I}_D$ is an identity matrix of size D.

The matrix $\mathbf{Z}$ is sampled in two steps: the first involves sampling existing features, and the second, sampling new features. The latent variables x_{kn} are marginalised since the collapsed Gibbs sampler can lead to faster convergence [89]; the marginal distribution is available in closed form as the Gaussian prior over the hidden sources is conjugate to the Gaussian likelihood over the observed data. The existing features z_{kn} can be sampled directly using the Bernoulli posterior:

$$z_{kn}|\mathbf{y}_n,\mathbf{x}_n,\mathbf{w}_k,\sigma_W^2,\sigma^2 \sim \text{Bernoulli}\left(\frac{\mathrm{P}\left(\mathbf{y}_n|z_{kn}=1\right)\mathrm{P}\left(z_{kn}=1|\mathbf{z}_{k-n}\right)}{\mathrm{P}\left(\mathbf{y}_n|z_{kn}=1\right)\mathrm{P}\left(z_{kn}=1|\mathbf{z}_{k-n}\right)+\mathrm{P}\left(\mathbf{y}_n|z_{kn}=0\right)\mathrm{P}\left(z_{kn}=0|\mathbf{z}_{k-n}\right)}\right),$$

where $\mathbf{z}_{k-n}$ is the kth row of the matrix $\mathbf{Z}$ without the nth element, $\mathrm{P}\left(z_{kn}=1|\mathbf{z}_{k-n}\right)=\frac{\sum_{i\neq n}z_{ki}}{N}$, and $\mathrm{P}\left(z_{kn}=0|\mathbf{z}_{k-n}\right)=\frac{N-\sum_{i\neq n}z_{ki}}{N}$. The posterior for new features are not available in closed form, but it can be approximated using a Metropolis-Hastings step. For each observation, adding κ number of new features and their corresponding parameters (columns of the matrix $\mathbf{W}$) are jointly proposed and accepted with probability proportional to likelihood improvement brought about by these new features; see equation (13) in [85].

5.4 Relevance determination with discrete variables

Latent class FA models make rigid partitioning assumptions, they fail to model richer clustering topologies, but latent feature FA models can infer uninterpretable features if not adequately constrained. For example, in situations when a beta-Bernoulli prior

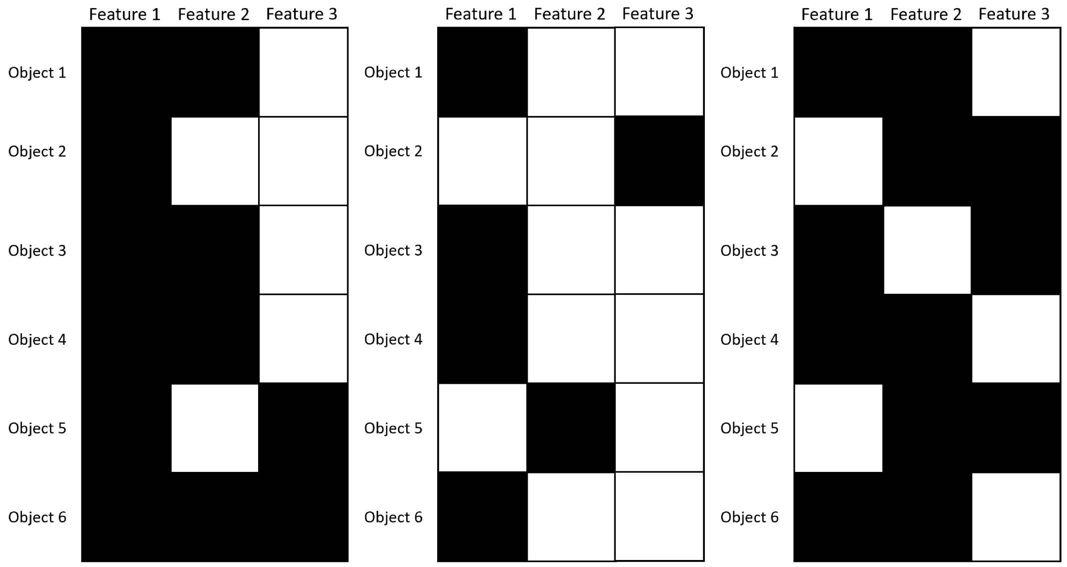


Figure 5.1: A plot of three different binary matrix: black cells indicate 1's and white cells indicate 0's. (a) Latent feature binary matrix, (b) Latent class binary matrix, and (c) Hypergeomeric (from equation (5.4)) binary matrix with $K = 3$ and $L = 2$.

is placed on the latent space, the number of data points being active in each feature follows *Zipf's law* [5, 90]; this implies that a small number of data points are active in all features, and a large number of data points are only active in a small number of features; [5] has proven this as $N \to \infty$. See Figure 5.1

The common 'rich-gets-richer' assumption underlying the feature allocation behaviour of many latent feature linear Gaussian models (see Section 5.3). Whereas latent feature FA models capture richer clustering topologies when compared to latent class FA models, beta-Bernoulli models (and many related constructions) impose strong assumptions about the distribution of the total number of factors and individual factor allocation frequency. Instead, a flexible relevance determination assumption can be that each input data point is associated with a different subset of L features, selected from a total of K unique features. The parameter K then accounts for the global sharing of structure across overlapping groups of data points with common features; if K is large enough, each point can be associated with non-overlapping subsets of L features. This behaviour is equivalent to the mixture of the FAs model. As K decreases, more of these features are constrained to be shared across subsets of the data. L acts much like the number of latent dimensions in traditional linear latent variable models, but here L is constrained by K. Thus, we can interpret L as the *local capacity* of the model, with K controlling the *global sharing capacity.* If $L = K$, we recover the classical linear Gaussian model of equation (5.1), since all features are associated with all observed data points. As $K - L$ increases, more local structures in the data can be represented.

If we assume that, for each column of $\mathbf{Z}$ in a latent feature model, exactly L out of K features are non-zero, then there are $\binom{K}{L}$ possible configurations for each column of $\mathbf{Z}$. When a flat prior is placed on $\mathbf{Z}$, then each configuration has an equal likelihood, $\frac{1}{\binom{K}{L}}$. The joint probability over any particular allocation matrix is given by the product:

$$\mathrm{P}\left(\mathbf{Z}|K, L\right) = \prod_{n=1}^{N} \frac{1}{\binom{K}{L}},$$

The restriction that L out of K features are non-zero, means that the rows of $\mathbf{Z}$ can no longer be distributed across the K feature columns. Instead, each of the $\frac{1}{\binom{K}{L}}$ configurations has a categorical likelihood that depends on a different combination of L non-zero factor loadings:

$$\mathrm{P}\left(\mathbf{z}_n = \mathbf{z}^* | \mathbf{y}_n, \mathbf{x}_n, \{\mathbf{w}_k\}_{k=1}^{K}, \sigma^2\right) \propto \prod_{j \in \mathbf{z}*} \mathrm{P}\left(\mathbf{y}_n | \mathbf{x}_n, \mathbf{w}_j, \sigma^2\right),$$

where $\mathbf{z}^*$ denotes some K-dimensional configuration $[1, 0, 0, 1, \ldots, 0, 1]^T$ with L 1's. For large K and small L, the number of unique configurations $\mathbf{z}^*$ grows factorially and we may wish to approximate the posterior over $\mathbf{z}_n$ assuming conditional independence of the feature assignments. In that case, we sequentially sample without replacement from the categorical posterior:

$$\begin{aligned} l | \mathbf{y}_n, \{\mathbf{w}_k\}_{k=1}^{K}, \mathbf{x}_n, \sigma^2 \sim \text{Categorical} \Bigg(& \frac{(1 - z_{1,n}) \, \mathrm{P}\left(\mathbf{y}_n | \mathbf{w}_1, \mathbf{z}_{-1,n}, \mathbf{x}_n, \sigma^2\right)}{\sum_{k=1}^{K} (1 - z_{k,n}) \, \mathrm{P}\left(\mathbf{y}_n | \mathbf{w}_k, \mathbf{z}_{-k,n}, \mathbf{x}_n, \sigma^2\right)}, \\ & \ldots, \frac{(1 - z_{K,n}) \, \mathrm{P}\left(\mathbf{y}_n | \mathbf{w}_K, \mathbf{z}_{-K,n}, \mathbf{x}_n, \sigma^2\right)}{\sum_{k=1}^{K} (1 - z_{k,n}) \, \mathrm{P}\left(\mathbf{y}_n | \mathbf{w}_k, \mathbf{z}_{-k,n}, \mathbf{x}_n, \sigma^2\right)} \Bigg). \end{aligned} \tag{5.4}$$

This information is encoded in $\mathbf{z}_n$ by setting the lth element to 1, this is repeated L times to ensure each $\mathbf{z}_n$ satisfies the constraint that L features are allocated per point. The posterior in equation (5.4) is identical to the posterior in a mixture model, where observation n is assigned to a class (mixture component) using the categorical distribution, except in this case we allow observation n to belong to L classes. If we are to further assume that the probability of each feature being active depends on how often it is selected in the rest of the data (i.e. the 'rich-get-richer' effect applies), we augment equation (5.4) with independent counts:

$$
\begin{aligned}
l|\mathbf{y}_n, \{\mathbf{w}_k\}_{k=1}^{K}, \mathbf{x}_n, \sigma^2 \sim \\
\text{Categorical}\Bigg(& \frac{(1 - z_{1,n})\, m_1^{(-n)} \mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_1, \mathbf{z}_{-1,n}, \mathbf{x}_n, \sigma^2\right)}{\sum_{k=1}^{K} (1 - z_{k,n})\, m_k^{(-n)} \mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \mathbf{z}_{-k,n}, \mathbf{x}_n, \sigma^2\right)}, \\
& \ldots, \frac{(1 - z_{K,n})\, m_K^{(-n)} \mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_K, \mathbf{z}_{-K,n}, \mathbf{x}_n, \sigma^2\right)}{\sum_{k=1}^{K} (1 - z_{k,n})\, m_k^{(-n)} \mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \mathbf{z}_{-k,n}, \mathbf{x}_n, \sigma^2\right)}\Bigg),
\end{aligned}
$$

where $m_k^{(-n)} = \sum_{i \neq n} z_{k,i}$. This results in a *multivariate hypergeometric* model for the numbers of active allocations in $\mathbf{Z}$.

Constrained factor allocation

Above, we assumed that the main feature allocation constraint is the number of non-zero factor loadings per point (i.e., a row-wise sparsity constraint on **Z** depending on L). However, we can also control the column-wise sparsity using constraints on the total number of times that a factor can be allocated to a point. Let us assume that each data point is associated with L out of K factors, and explicitly model the number of non-zero factor loadings across columns. We can place the *truncated multinomial distribution* (see equation (5.5)) as a prior on **Z** with K different categories and probability of success π_k for $k = 1, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$. The truncated multinomial can be used to restrict the number of trials L, as well as the total number of times a category can be selected, i.e. c_k for each category k with $\sum_{k=1}^{K} c_k \geq L$. A sample from the truncated multinomial distribution is then a K-dimensional vector $\mathbf{m}$ of counts:

$$
\mathrm{P}\left(\mathbf{m}|\boldsymbol{\pi}\right) = \frac{L!}{V\left(\boldsymbol{\pi}, L, \mathbf{c}\right)} \prod_{k=1}^{K} \frac{\pi_k^{m_k}}{m_k!}, \tag{5.5}
$$

where $\mathbf{m} = [m_1, \ldots, m_K]^T$ and $V(\boldsymbol{\pi}, L, \mathbf{c})$ is a normalising constant:

$$V(\boldsymbol{\pi}, L, \mathbf{c}) = \sum_{l_1=0}^{c_1} \cdots \sum_{l_K=0}^{c_K} \left(\mathbb{I}\left(\sum_i l_i = L \right) L! \prod_{k=1}^{K} \frac{\pi_k^{l_k}}{l_k!} \right),$$

where $\mathbb{I}(\cdot)$ is an indicator function which is one if the statement inside is true, otherwise zero. Under this constrained model, we can write the conditional probability over the sparse matrix **Z** given $\pi = [\pi_1, \ldots, \pi_K]$:

$$\mathrm{P}(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{P}(z_{nk}|\pi_k, \mathbf{c}^*) \propto \prod_{n=1}^{N} \frac{L!}{V(\boldsymbol{\pi}, L, \mathbf{c})} \prod_{k=1}^{K} \frac{\pi_k^{z_{nk}}}{z_{nk}!}, \tag{5.6}$$

where $\mathrm{P}(z_{nk}|\pi_k, \mathbf{c}^*)$ cannot be easily distributed since we need to keep track of $\mathbf{c}^*$ the total number of available draws from each factor, such that $\sum_{k=1}^{K} z_{nk} = L$. In the fully Bayesian setting, one can model the allocation marginal probabilities $\pi_1, \ldots, \pi_K$ with a Dirichlet distribution parametrised by the counts $m_1, \ldots, m_K$. A nonparametric extension of this constrained model can be derived by taking the limit $K \to \infty$ in equation (5.6) and integrating out π.

5.4.1 Adaptive FA

The *adaptive* FA (aFA) [4] model also deviates from the data generation process in equation (5.3) in three ways: (1) It assumes a multivariate hypergeometric prior on the latent variables **Z** with parameter L, (2) The model noise has a single variance parameter σ^2 (i,e., $\boldsymbol{\epsilon}_n \sim \mathcal{MVN}(\mathbf{0}, \sigma^2 \mathbf{I}_D)$, and (3) The factor loading variables have fixed variance $\sigma_w^2 = 1$ (i.e., $\mathbf{w}_k \sim \mathcal{MVN}(\mathbf{0}, \mathbf{I}_D)$).

Inference

The data log-likelihood for the proposed aFA model is:

$$\mathcal{L}_N = -\sum_{n=1}^{N} \Bigg(\frac{K}{2} \ln\left(\sigma_x^2\right) + \frac{D}{2} \ln\left(\sigma^2\right) \\ + \frac{1}{2\sigma_x^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_n + \frac{1}{2\sigma^2} \mathbf{y}_n^{\mathrm{T}} \mathbf{y}_n - \frac{1}{\sigma^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{y}_n + \frac{1}{2\sigma^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n \mathbf{x}_n \Bigg),$$

where $\mathbf{A}_n = \operatorname{diag}\left(\mathbf{z}_n\right)$ is a $(K \times K)$ matrix with the diagonal elements set to $\mathbf{z}_n$. The parametric nature of the hypergeometric distribution over $\mathbf{Z}$ allows for an efficient *expectation-maximisation* (EM) algorithm for training the aFA model, which can be used both for initialisation of a full Gibbs sampler or for rapidly obtaining a (local) maximum-a-posteriori solution for the model; however, the factor loading matrix $\mathbf{W}$ is now treated as a parameter instead of a random variable. The EM scheme iteratively switches between taking the expectation step (E-step) and the maximisation step (M-step). In the E-step, the expectation of the latent variables $\{\mathbf{x}_n\}_{n=1}^{N}$ is taken with respect to the posterior distribution; this results in the complete data log-likelihood:

$$\mathcal{L}_N^{\text{complete}} = -\sum_{n=1}^{N} \Bigg(\frac{K}{2} \ln\left(\sigma_x^2\right) + \frac{D}{2} \ln\left(\sigma^2\right) \\ + \frac{1}{2\sigma_x^2} \operatorname{tr}\left(\mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}\right]\right) + \frac{1}{2\sigma^2} \mathbf{y}_n^{\mathrm{T}} \mathbf{y}_n - \frac{1}{\sigma^2} \mathbb{E}\left[\mathbf{x}_n\right]^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{y}_n \\ + \frac{1}{2\sigma^2} \operatorname{tr}\left(\mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n \mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}\right]\right) \Bigg), \tag{5.7}$$

where:

$$\mathbb{E}\left[\mathbf{x}_n\right] = \left(\sigma_x^{-2} \mathbf{I}_K + \sigma^{-2} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n\right)^{-1} \left(\sigma^{-2} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{y}_n\right),$$

$$\mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}\right] = \left(\sigma_x^{-2} \mathbf{I}_K + \sigma^{-2} \mathbf{A}_n \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n\right)^{-1} + \mathbb{E}\left[\mathbf{x}_n\right] \mathbb{E}\left[\mathbf{x}_n\right]^{T}.$$

In the M-step, the complete data log-likelihood in equation (5.7) is maximised with respect the other parameters; this is done by solving the following differential equations

$\frac{\partial \mathcal{L}_N^{\text{complete}}}{\partial \mathbf{W}} = 0$, $\frac{\partial \mathcal{L}_N^{\text{complete}}}{\partial \sigma_x} = 0$ and $\frac{\partial \mathcal{L}_N^{\text{complete}}}{\partial \sigma} = 0$, which results in the following updates:

$$
\begin{aligned}
\mathbf{W} &= \left(\sum_{n=1}^{N} \mathbf{y}_n \left(\mathbf{A}_n \mathbf{x}_n \right)^{\mathrm{T}} \right) \left(\sum_{n=1}^{N} \mathbf{A}_n \mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \right] \mathbf{A}_n \right)^{-1}, \\
\sigma^2 =& \frac{1}{ND} \sum_{n=1}^{N} \left(\mathbf{y}_n^{\mathrm{T}} \mathbf{y}_n - 2\mathbf{x}_n^{T} \mathbf{A}_n \mathbf{W}^{\mathrm{T}} \mathbf{y}_n + \operatorname{trace}\left(\mathbf{A}_n \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n \mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \right] \right) \right), \\
\sigma_x^2 =& \frac{1}{NK} \sum_{n=1}^{N} \operatorname{trace}\left(\mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \right] \right).
\end{aligned}
$$

Each latent variable $\mathbf{z}_n$ has L indices $\{l_1, \ldots, l_L\}$ for which $z_{l_i n} = 1$; therefore the M-step update for the latent variable $\mathbf{z}_n$ boils down to finding the L indices $\{l_1, \ldots, l_L\}$ for each observation n which maximises the complete log-likelihood in equation (5.7); this is highlighted in Algorithm 3. Since we are often interested only in a point estimate for the indicator variables $\mathbf{Z}$, iterative optimisation via coordinate descent can lead to a robust, local MAP estimate i.e. $\mathbf{Z}^{\text{MAP}}$ [62, 91–93]. The complete EM algorithm for the proposed aFA is summarised in Algorithm 4; with $\boldsymbol{\epsilon}_{-kn} = \mathbf{y}_n - \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n \right)$ with $z_{kn} = 0$, or the noise associated with nth point and kth feature.

Algorithm 3 Update latent varibles $\{\mathbf{z}_n\}_{n=1}^{N}$

```
for $n \leftarrow 1$ to $N$
    Set $\mathbf{z}_n = \mathbf{0}$
    for $i \leftarrow 1$ to $L$
        Set $l_i = \text{argmax}\Bigg( (1 - z_{1n}) \frac{\sigma^2}{\sigma^2 + \mathbf{w}_1^T \mathbf{w}_1} \exp\left( 0.5 \left( \frac{\mathbf{w}_1^T \boldsymbol{\epsilon}_{-1n}}{\sigma^2} \right)^2 \right), \ldots$
            $\ldots (1 - z_{Kn}) \frac{\sigma^2}{\sigma^2 + \mathbf{w}_K^T \mathbf{w}_K} \exp\left( 0.5 \left( \frac{\mathbf{w}_K^T \boldsymbol{\epsilon}_{-Kn}}{\sigma^2} \right)^2 \right) \Bigg)$
        Set $z_{l_i n} = 1$
```

5.4.2 Experiments

Synthetic data

Five different data sets are generates, each with follow the generative process described in equation (5.3); with without the loss of generality $\sigma_w^2 = 1$, $\boldsymbol{\sigma}^2 = \sigma^2 = 0.1^2$ and $\sigma_x^2 = 1$. The core of the generative model remains the same across the different data sets we generate ($N = 1200$, $D = 35$), and only the number of latent features K and

Algorithm 4 EM pseudocode for parametric adaptive factor (aFA) analysis.

```
Input: $\mathbf{Y}, \boldsymbol{\Theta}, \text{MaxIter}$
Initialise: Sample a random $(K \times N)$ binary matrix $\mathbf{Z}$ and initialize $\{\mathbf{W}, \mathbf{X}\}$ using PCA
for iter $\leftarrow 1$ to MaxIter
    for $n \leftarrow 1$ to $N$
        Set $\mathbf{x}_n = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1}\left(\sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{y}_n\right)$
        Set $\boldsymbol{\Psi}_n = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1} + \mathbf{x}_n\mathbf{x}_n^T$
    Set $\{\mathbf{z}_n\}_{n=1}^N$ using Algorithm 3
    Set $\mathbf{W} = \left(\sum_{n=1}^N \mathbf{y}_n\left(\mathbf{A}_n\mathbf{x}_n\right)^{\mathrm{T}}\right)\left(\sum_{n=1}^N \mathbf{A}_n\boldsymbol{\Psi}_n\mathbf{A}_n\right)^{-1}$
    Set $\sigma^2 = \frac{1}{ND}\sum_{n=1}^N\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{y}_n - 2\mathbf{x}_n^T\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{y}_n + \text{trace}\left(\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\boldsymbol{\Psi}_n\right)\right)$
    Set $\sigma_x^2 = \frac{1}{NK}\sum_{n=1}^N \text{trace}\left(\boldsymbol{\Psi}_n\right)$
```

the prior distribution $\mathcal{F}(\cdot)$ on the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$ changes. Samples from five different prior distributions $\mathcal{F}(\cdot)$ are displayed in Figure 5.2. Five different FA methods (i.e. with changing treatment of $\{\mathbf{z}_n\}_{n=1}^N$) are compared on the different data sets; each model was trained on 80% of the data set and tested on the remaining 20%; $K = 10$ results are reported in Table 5.1, and $K = 20$ results are reported in Table 5.2. The five FA methods tested are:

- Factor analysis (FA): no binary latent variables $\mathbf{z}_{1,\dots,N}$ exists; see Section 5.2.
- Infinite sparse FA (isFA): an IBP prior is placed on the binary latent variables $\mathbf{z}_{1,\dots,N}$; see Section 5.3.1.
- Finite sparse FA (fsFA): the binary latent variables $\mathbf{z}_{1,\dots,N}$ are modelled with a finite Beta-Bernoulli distribution across all points and features; in other words a truncated version of isFA described in Section 5.3.1.
- Adaptive FA (aFA): a multivariate hypergeometric prior is placed in the binary latent variables $\mathbf{z}_{1,\dots,N}$; see Section 5.4.1.
- Sparse and dense FA (sdFA) [88]: the factor loading matrix $\mathbf{W}$ is split into two components; one component is dense and the other component is sparse.

Both Table 5.1 and Table 5.2 also include a second result for the aFA model when trained using the proposed EM algorithm (Algorithm 4). This was done to distinguish between performance gains due to the model architecture and due to inference method. The

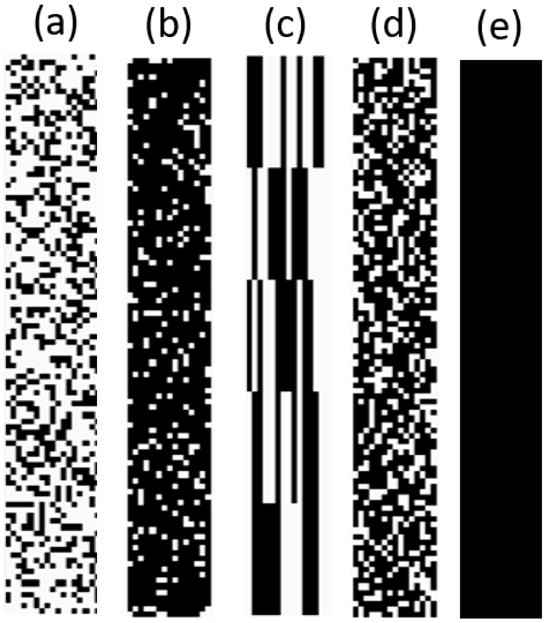


Figure 5.2: A plot of the different distributions used to model the latent space in Table 5.1 and Table 5.2. The subplots display different samples of the binary latent variables $\mathbf{z}_{1,\ldots,N}$: black cells indicate 1's and white cells indicate 0's. Five different latent models are considered: (a) Sparse latent feature model, (b) Dense latent feature model, (c) Latent class model in which sharing of some feature between subsets of points implies sharing of all features of those points, (d) Balanced latent feature model sampled from specific hypergeometric distribution, (e) Collapsed latent space consisting of a single state.

results in Table 5.1 and Table 5.2 suggest that for sparse latent feature data and for single feature linear Gaussian data, most of the methods perform similarly. The isFA model performs consistently worse than all other methods due to its tendency to overestimate the underlying number of latent features. When we set the concentration parameters of isFA to learn the fixed K number of factors, reconstruction error is higher; if we set concentration parameters to infer the number of factors that is higher than the true generating number of factors K, the reconstruction error drops. This effect is similar to the one reported by [94] for Dirichlet process mixtures. Likewise, the sdFA model also performs poorly across all data sets; this is probably because the sdFA assumes that all sparsity can be modelled using the factor loading matrix.

FA performs well in terms of reconstruction error since it uses all factors to express all points, i.e., it learns a lot more loadings than the alternative models with a more parsimonious structure. In practice, latent feature FA methods are used with larger K than FA since for each factor there is a linear combination of only a small subset of data points. The fsFA manages to perform well across most settings, often achieving comparable reconstruction error using a lot of sparser factor loadings. However, we see its performance drop substantially for non-sparse balanced latent feature models. Due to the generality of the aFA model, it performs well across all settings since the latent space structures in the synthetic data are all special cases for the multivariate hypergeometric model. The slightly lower reconstruction error of EM versus Gibbs aFA, suggests convergence to good local optima for the proposed EM scheme and convergence issues of the Gibbs sampler.

Factor sharing between different digits on the MNIST data set

In this section, we demonstrate training the proposed aFA model on $N = 2500$ odd-labelled digits (500 of each type) from the MNIST handwritten digit data set [95]. The raw pixel data were first reduced to $D = 350$ using standard PCA since this still preserves 99.5% of the total variance within the data. The total number of unique factors is set to $K = 100$ and the number L of observation-specific factors is set to maximise

Table 5.1: Performance of different variations of factor analysis (FA) methods on five different data sets of dimension $D = 35$ and latent features $K = 10$; the data generating process is described in equation (5.3). Each model was trained on 80% of the data set and tested on the remaining 20%, the average mean squared error and one standard deviation from 20 different experiments are reported.

Prior	Sparse matrix	Balance matrix	Dense matrix	Single state	Subspace clustering
Factor analysis (FA)	0.260 ±0.01	0.269 ±0.01	0.265 ±0.02	0.292 ±0.02	0.272 ±0.02
Finite sparse FA	0.347 ±0.03	0.349 ±0.02	0.349 ±0.03	0.348 ±0.02	0.348 ±0.03
Infinite sparse FA	0.351 ±0.05	0.395 ±0.09	0.361 ±0.14	0.357 ±0.15	0.363 ±0.06
Adaptive FA (aFA) Gibbs	0.343 ±0.03	0.346 ±0.02	0.346 ±0.03	0.347 ±0.02	0.345 ±0.02
Adaptive FA (aFA) EM	**0.247** **±0.00**	**0.251** **±0.01**	**0.250** **±0.01**	**0.251** **±0.02**	**0.248** **±0.01**
Sparse & dense FA	0.757 ±0.15	0.801 ±0.18	0.901 ±0.02	1.01 ±0.10	0.712 ±0.05

Table 5.2: Performance of different variations of factor analysis (FA) methods on five different data sets of dimension $D = 35$ and latent features $K = 20$; the data generating process is described in equation (5.3). Each model was trained on 80% of the data set and tested on the remaining 20%, the average mean squared error and one standard deviation from 20 different experiments are reported.

Prior	Sparse matrix	Balance matrix	Dense matrix	Single state	Subspace clustering
Factor analysis (FA)	0.159 ±0.01	0.162 ±0.01	0.165 ±0.01	0.169 ±0.01	0.162 ±0.01
Finite sparse FA	0.340 ±0.03	0.340 ±0.03	0.341 ±0.03	0.342 ±0.03	0.341 ±0.02
Infinite sparse FA	0.336 ±0.04	0.337 ±0.04	0.340 ±0.03	0.343 ±0.02	0.344 ±0.02
Adaptive FA (aFA) Gibbs	0.335 ±0.02	0.336 ±0.02	0.338 ±0.03	0.339 ±0.03	0.342 ±0.02
Adaptive FA (aFA) EM	**0.136** **±0.00**	**0.136** **±0.00**	**0.140** **±0.01**	**0.140** **±0.01**	**0.139** **±0.01**
Sparse & dense FA	0.482 ±0.04	0.512 ±0.10	0.756 ±0.04	0.577 ±0.09	0.521 ±0.02

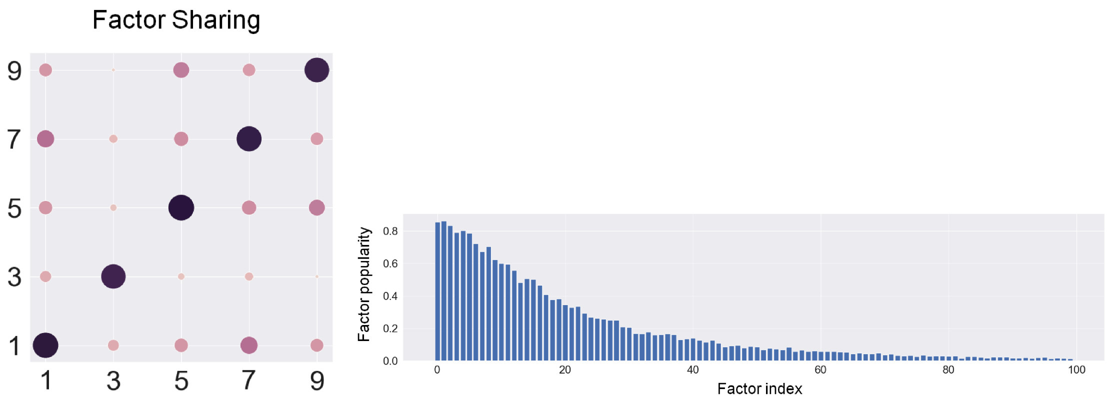


Figure 5.3: aFA model trained on 2500 odd-labelled MNIST digits, 500 of each label. **Left**: Factor sharing grid between digits: circles are sized depending on the number of features shared between digit pairs denoted on the x-axis and y-axis; colour enforces this effect where darker circles indicate more sharing and brighter circles - less. **Right**: Distribution of feature allocation processes: y-axis denotes the proportion of data sharing the current factor; x-axis indicates the factor number where the factors are ordered based on most popular (left), to least popular with a small number of data points allocated.

the factor profiles of the different digits.

In Figure 5.3, we show the factor sharing across the digits which are calculated based on the proportion of factors shared between different digit pairs observations. We count the number of factors shared between samples of 1's and 1's, 1's and 3's, 1's and 5's, 1's and 7's, 1's and 9's, then we normalise by the largest number of features shared; the procedure is repeated for the full grid. The larger and darker circles indicate sharing of more factors. As expected, observations depicting the same digits have the most shared factors; 1's and 7's also share significant structure as well as 5's and 9's which broadly coincide with the geometry of the digits. The results can be directly compared with a similar experiment in [96]. In Figure 5.3 we display the estimated feature weights obtained by summing over the latent variables $\mathbf{z}_{1,\dots,N}$ and normalising. Varying L and K one can study how well sparse and dense aFA models infer features specific to the different digits.

5.5 Principal component analysis

Principal component analysis (PCA) is still one of the most common (and oldest) methods used in dimensionality reduction [97]. PCA uses an orthogonal transformation to transform observed data into linearly uncorrelated principal components. Let $\{\mathbf{y}_n\}_{n=1}^{N}$ be N D-dimensional data points and let $\{\mathbf{w}_k\}_{k=1}^{K}$ be K orthonormal D-dimensional principal component loadings; where the orthonormality results in $\mathbf{w}_i^T \mathbf{w}_j = 0$ for $i \neq j$. The vectors $\{\mathbf{w}_k\}_{k=1}^{K}$ are the K largest (with respect to eigenvalues) eigenvectors of the sample covariance matrix $\mathbf{\Sigma} = \frac{1}{N}\sum_n^N (\mathbf{y}_n - \boldsymbol{\mu})^T (\mathbf{y}_n - \boldsymbol{\mu})$ where $\boldsymbol{\mu} = \frac{1}{N}\sum_n^N y_n$ is the sample mean. Then the K principal component scores of the data point $\mathbf{y}_n$ are $\mathbf{x}_n = \mathbf{W}^T (\mathbf{y}_n - \boldsymbol{\mu})$; alternatively $\mathbf{x}_n$ can be viewed as the 'lower K-dimensional representation' of $\mathbf{y}_n$.

5.5.1 Probabilistic PCA

The probabilistic PCA [98] uses the linear Gaussian LVM form from equation (5.1) in the following way:

$$\begin{aligned}
\boldsymbol{\epsilon}_n|\sigma^2 &\sim \mathcal{MVN}(0, \sigma^2 \mathbf{I}_D), \\
\mathbf{x}_n &\sim \mathcal{N}(0, \mathbf{I}_K), \\
\mathbf{y}_n &= \mathbf{W}\mathbf{x}_n + \boldsymbol{\epsilon}_n.
\end{aligned}$$

To infer the random variables $\mathbf{x}_n$ and the parameters $\{\mathbf{W}, \sigma\}$, [98] propose a EM algorithm which first takes the expectation (E-step) of the latent variables $\mathbf{x}_n$ with respect to the posterior distribution and then maximises (M-step) the complete data log-likelihood with respect to the parameters $\{\mathbf{W}, \sigma\}$.

An alternative way to infer the parameters $\{\mathbf{W}, \sigma\}$ is to first marginalise out the latent variables $\mathbf{x}_n$, and then maximise the marginal likelihood with respect to the parameters. The maximum marginal likelihood update for the parameter $\mathbf{W}$ is the K largest (with respect to eigenvalues) eigenvectors of the sample covariance matrix $\mathbf{\Sigma} = \frac{1}{N}\sum_n^N (\mathbf{y_n} - \boldsymbol{\mu})^T (\mathbf{y_n} - \boldsymbol{\mu})$. Hence, the orthonormal $\mathbf{W}$ gives a rational behind the name principal. Furthermore, the PCA can be obtained by assuming a small variance asymptotic (SVA) of $\sigma \to 0$.

5.6 Latent variable probabilistic principal component analysis models

Latent feature visualisation counterparts have received a lot less attention, despite the popularity of sparse principal component analysis techniques [99, 100]. This is most likely due to the complexity of specifying distributions over orthogonal matrices and the difficulty of performing inference with them. In this section, we extend the latent factor analysis work (see Section 5.3) into the PPCA setup:

$$\begin{aligned}
\mathbf{W} &\sim \mathcal{B}(\cdot), \\
\boldsymbol{\epsilon}_n|\sigma^2 &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2\mathbf{I}_D\right), \\
\mathbf{x}_n &\sim \mathcal{MVN}\left(\mathbf{0}, \mathbf{I}_K\right), \\
\mathbf{z}_n &\sim \mathcal{F}(\cdot), \\
\mathbf{y}_n &= \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n,
\end{aligned} \tag{5.8}$$

where $\mathcal{B}(\cdot)$ is some distribution which samples the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ with orthonormal columns; i.e. $\mathbf{w}_i^T\mathbf{w}_j = 0$ if $i \neq j$ and $\mathbf{w}_i^T\mathbf{w}_i = 1$, $\mathbf{z}_n$ is a K dimensional binary latent variable sampled from some distribution $\mathcal{F}(\cdot)$, $\odot$ denotes the Hadamard product, and $\mathbf{I}_D$ is a identity matrix of size D. The generative process in equation (5.8) and equation (5.3) only differ in their assumption on the projection matrix $\mathbf{W}$. In this section two different variants of latent variable PPCA are introduced; each assuming different prior $\mathcal{F}(\cdot)$ on the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$. The first variant *infinite sparse* PPCA (isPPCA), places an IBP prior on the binary latent variables. The second variant is the *adaptive* PPCA (aPPCA) [4], this places a multivariate hypergeometric prior on the binary latent variables.

As discussed in Section 5.5.1, the principal component name comes from the fact that the maximum likelihood (marginal) update for the parameter $\mathbf{W}$ is the K largest (with respect to eigenvalues) eigenvectors the sample covariance matrix. However, the different priors $\mathcal{F}(\cdot)$ on the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$ result in explicit control over the scale of the different projection axis; this can address well known pitfalls of PCA such as the disproportionate crowding of the projections due to outliers or multi-modalities and the sphericalisation of the projection; see Appendix C.1.

Inference

Computing the posterior distribution of the latent variables $\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N$ and the projection matrix $\mathbf{W}$ is analytically intractable and we have to resort to approximate inference. Unlike for aFA from Section 5.4.1, the posterior updates of the orthonormal matrix $\mathbf{W}$ do not allow for closed form updates. At the same time numerically optimising over $\mathbf{W}$ and marginalizing $\{\mathbf{x}\}_{n=1}^N$ leads to slow mixing and an EM scheme leads to poor local solutions for this model. An efficient Markov Chain Monte Carlo (MCMC) scheme [101] can be derived which iterates between explicit updates for $\mathbf{W}$, $\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N$, and others. Sampling from directional posteriors is prohibitively slow, so we propose a MAP scheme for the updates on $\mathbf{W}$. Alternatively, we could use an automated MCMC platforms such as STAN [102] for the inference, but STAN does not deal well with discontinuous likelihood models introduced by the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$. This can be addressed using discrete relaxations such as [35] or numerical solver extensions such as [103]. However, such an approach can be justified only for nonlinear intractable extensions of latent feature PPCA, since the Gibbs sampler with closed-form updates is substantially more efficient.

The joint data likelihood of both latent feature subspace models we propose takes the form:

$$
\begin{aligned}
\mathrm{P}\left(\mathbf{Y},\mathbf{W},\mathbf{X},\mathbf{Z}|\sigma,\boldsymbol{\eta}\right) = \prod_{n=1}^{N}&\left(\mathrm{P}\left(\mathbf{y}_n|\mathbf{W},\mathbf{x}_n,\mathbf{z}_n,\sigma\right)\prod_{k=1}^{K}\mathrm{P}\left(x_{kn}\right)\mathrm{P}\left(z_{kn}|\boldsymbol{\eta}\right)\right) \\
&\times \mathrm{P}\left(\mathbf{W}\right),
\end{aligned}
\tag{5.9}
$$

where isPPCA model and the aPPCA differ on the prior on the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$, hence in the aPPCA model $\boldsymbol{\eta} = \{L, K\}$, and in the isPPCA model $\boldsymbol{\eta} = \{\alpha\}$. We can check whether the MCMC sampler has converged using standard tests such as [104] directly on equation (5.9). The inference of both models only differs with updating the binary latent variables $\{\mathbf{z}_n\}_{n=1}^N$, therefore the posterior updates on the other parameters will be introduced together for both models.

Posterior of W

It is important to use a distribution with support on the Stiefel manifold (see [105] for a good introduction) to comply with the orthonormal constraint on $\mathbf{W}$. [106] explored exactly this problem in the context of latent feature subspace modelling and proposed using a conjugate *Bingham* prior [107] independently on the columns of $\mathbf{W}$ leading to an independent *von Mises-Fisher* posterior over each column where re-scaling is required after each sample to maintain orthonormal. However, empirical trials suggest that this results in very poor mixing (see Appendix C.2). To overcome this issue, we propose joint sampling of the columns of $\mathbf{W}$. We place a uniform prior over the Stiefel manifold on the matrix $\mathbf{W}$ which allows us to work with a *matrix von Mises-Fisher* [108] posterior:

$$
\mathrm{P}\left(\mathbf{W}|\mathbf{Y},\mathbf{X},\mathbf{Z},\sigma\right) = {}_0\mathrm{F}_1^{-1}\left(\emptyset,\frac{D}{2},\mathbf{B}\mathbf{B}^T\right)\exp\left(\mathrm{tr}\left(\mathbf{B}\mathbf{W}\right)\right), \tag{5.10}
$$

where $\mathbf{Y} = [\mathbf{y}_1,\ldots,\mathbf{y}_N]$, $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_N]$, $\mathbf{Z} = [\mathbf{z}_1,\ldots,\mathbf{z}_N]$, $\mathbf{B} = \frac{1}{2\sigma^2}\left(\mathbf{X}\odot\mathbf{Z}\right)\mathbf{Y}^T$ and ${}_0\mathrm{F}_1^{-1}\left(\cdot\right)$ is a hypergeometric function [109]. The normalisation term of the matrix von Mises-Fisher posterior is not available in closed form, hence it is common to sample from it using rejection sampling. [110] proposed a Metropolis-Hastings scheme to generate samples from equation (5.10), the resulting posterior of $\mathbf{W}$ converges faster than the

Bingham-von-Mises-Fisher posterior, but can be further sped up by numerical optimisation methods. Here, we propose updating the matrix **W** by maximising the posterior from equation (5.10) over the Stiefel manifold, i.e., keeping the orthonormality constrain $\mathbf{w}_i^T \mathbf{w}_j = 0$ if $i \neq j$ and $\mathbf{w}_i^T \mathbf{w}_i = 1$. An efficient implementation can be achieved using the PYMANOPT toolbox [111], for optimisation over manifolds with different geometries; this step is outlined in Appendix C.3.

Posterior of $\{\mathbf{x}_n\}_{n=1}^N$

The posterior distribution over the latent variable x_{kn}, for which its respective $z_{kn} = 1$, is sampled from a Gaussian:

$$x_{kn} | \mathbf{y}_n, \mathbf{w}_k, \sigma \sim \mathcal{N}\left(\frac{\mathbf{y}_n^T \mathbf{w}_k}{\sigma^2 + 1}, \frac{\sigma^2}{\sigma^2 + 1}\right) \tag{5.11}$$

where $\mathbf{w}_k$ is the kth column of the matrix **W**.

Posterior of σ^2

A inverse-Gamma prior is placed on σ^2 with parameters $\{\gamma, \vartheta\}$:

$$\mathrm{P}\left(\sigma^2 | \gamma, \vartheta\right) = \frac{\vartheta^\gamma}{\Gamma(\gamma)} \left(\sigma^2\right)^{-\gamma-1} \exp\left[-\frac{\vartheta}{(\sigma^2)}\right],$$

This leads to posterior distribution over σ^2 of the form:

$$\begin{aligned}
\mathrm{P}\big(\sigma^2 | \gamma, \vartheta, \mathbf{Y}, \mathbf{W}, \mathbf{X}, \mathbf{Z}\big) &= \frac{\vartheta^\gamma}{\Gamma(\gamma)} \left(\sigma^2\right)^{-\gamma-1} \exp\left[-\frac{\vartheta}{\sigma^2}\right] \times \frac{1}{\left(2\pi\sigma^2\right)^{\frac{ND}{2}}} \\
&\times \exp\left(-\frac{1}{2\sigma^2} \sum_{n=1}^N \left[\left(\mathbf{y}_n - \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right)\right)^T \left(\mathbf{y}_n - \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right)\right)\right]\right) \\
&\propto \left(\sigma^2\right)^{-(\gamma + ND/2) - 1} \\
&\times \exp\left(-\frac{1}{\sigma^2}\left(\frac{1}{2}\mathrm{tr}\left[\left(\mathbf{Y} - \mathbf{W}\left(\mathbf{X} \odot \mathbf{Z}\right)\right)^T \left(\mathbf{Y} - \mathbf{W}\left(\mathbf{X} \odot \mathbf{Z}\right)\right)\right] + \vartheta\right)\right),
\end{aligned} \tag{5.12}$$

which is still an inverse-Gamma distribution with parameters $\gamma^{post} = \gamma + \frac{ND}{2}$ and $\vartheta^{post} = \frac{1}{2}\mathrm{tr}\left[\left(\mathbf{Y} - \mathbf{W}\left(\mathbf{X} \odot \mathbf{Z}\right)\right)^T \left(\mathbf{Y} - \mathbf{W}\left(\mathbf{X} \odot \mathbf{Z}\right)\right)\right] + \vartheta$.

5.6.1 Infinite sparse PPCA

In the infinite sparse PPCA, an IBP prior is placed over the indicator matrix $\mathbf{z}_{1,\ldots,N}$; this assumes that after a finite N number of observations only a finite K number of one-dimensional subspaces are active. This results in the first K rows of $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ having non-zero entries, and the remaining being all zeros. By design, K cannot exceed the dimension of the data D and this leads to truncation of the IBP such that K has a upper limit of $K^{\max}$; where $K \leq K^{\max} \leq D$, therefore in the isPPCA, $\mathbf{Z}$ is a $(K^{\max} \times N)$ binary matrix, with the sum of the first K rows being non-zero and the sum of the remaining $K^{\max} - K$ rows being zero. We sample the matrix $\mathbf{Z}$ in two stages which include sampling 'existing features' and 'new features'; in both cases the latent variables x_{kn} are marginalized out. The posterior distribution over the existing features z_{kn} is Bernoulli distributed:

$$\begin{aligned}
&z_{kn}|\mathbf{y}_n, \mathbf{x}_n, \mathbf{w}_k, \sigma^2 \sim \\
&\qquad \text{Bernoulli}\left(\frac{\mathrm{P}\left(\mathbf{y}_n|z_{kn}=1\right)\mathrm{P}\left(z_{kn}=1|\mathbf{z}_{k-n}\right)}{\boldsymbol{\eta}}\right) \\
&\qquad = \text{Bernoulli}\left(\frac{\frac{m_{k-n}}{N}\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\left(\mathbf{y}_n^T\mathbf{w}_k\right)\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{1}{2}}}{\frac{m_{k-n}}{N}\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\left(\mathbf{y}_n^T\mathbf{w}_k\right)\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{1}{2}}+1}\right),
\end{aligned} \tag{5.13}$$

where $\boldsymbol{\eta} = \mathrm{P}\left(\mathbf{y}_n|z_{kn}=1\right)\mathrm{P}\left(z_{kn}=1|\mathbf{z}_{k-n}\right) + \mathrm{P}\left(\mathbf{y}_n|z_{kn}=0\right)\mathrm{P}\left(z_{kn}=0|\mathbf{z}_{k-n}\right)$, and $m_{k-n} = \sum_{i\neq n} z_{ki}$.

Then, we sample κ number of new features with $\kappa \sim$ Poisson $\left(\frac{\alpha}{N}\right)$, where we maintain $\kappa > 0$ or $\kappa + K \leq K^{\max}$. For observed data point n, the posterior distribution over the new features are:

$$\begin{aligned}
&z_{K+j,n}|\mathbf{y}_n, \mathbf{x}_n, \mathbf{w}_k, \sigma^2 \sim \\
&\qquad \text{Bernoulli}\left(\frac{\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\sum_{k=K+1}^{K+\kappa}\left(\mathbf{y}_n^T\mathbf{w}_k\right)^2\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{\kappa}{2}}}{\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\sum_{k=K+1}^{K+\kappa}\left(\mathbf{y}_n^T\mathbf{w}_k\right)^2\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{\kappa}{2}}+1}\right),
\end{aligned} \tag{5.14}$$

for $j = 1, \ldots, \kappa$ new features.

A Gamma prior is placed on the IBP concentration parameter α with parameters

$\{\lambda, \mu\}$:

$$\mathrm{P}\left(\alpha|\lambda,\mu\right) = \frac{\mu^{\lambda}}{\Gamma\left(\lambda\right)}\left(\alpha\right)^{\lambda-1}\exp\left(-\mu\alpha\right),$$

which results in the following posterior distribution:

$$\begin{aligned}\mathrm{P}\left(\alpha|\lambda,\mu,\mathbf{Y},\mathbf{W},\mathbf{X},\mathbf{Z}\right) &= \frac{\mu^{\lambda}}{\Gamma\left(\lambda\right)}\left(\alpha\right)^{\lambda-1}\exp\left[-\mu\alpha\right] \\ &\times \exp\left(-\alpha H_N\right)\alpha^{K}\times\left(\prod_{k=1}^{K}\frac{\left(m_k-1\right)!\left(N-m_k\right)!}{\left(N\right)!}\right) \\ &\propto \left(\alpha\right)^{\lambda+K-1}\exp\left(-\alpha\left(H_N+\mu\right)\right),\end{aligned} \tag{5.15}$$

which is still a gamma distribution with parameters $\lambda^{post} = \lambda + K$, $\mu^{post} = H_N + \mu$ and $H_N = \sum_{n=1}^{N}\frac{1}{n}$. The complete algorithm for the proposed isPPCA is summarised in Algorithm 5.

Algorithm 5 Gibbs sampling pseudocode for isPPCA.

```
Input: $\mathbf{Y}, \mathbf{\Theta}$, MaxIter, $K^{max}$
Initialise: Sample a random $(K^{max} \times N)$ binary matrix $\mathbf{Z}$ and initialize $\mathbf{W}$ using PCA
for iter ← 1 to MaxIter
  for $n$ ← 1 to $N$
    for $k$ ← 1 to $K$
      Sample $z_{kn}$ using equation (5.13)
    Sample $\kappa \sim$ Poisson $\left(\frac{\alpha}{N}\right)$
    Accept $\kappa$ new features with probability from equation (5.14) and update $K$ accordingly
  for $n$ ← 1 to $N$
    for $k$ ← 1 to $K$
      if $z_{kn} = 1$
        Sample $x_{kn}$ using equation (5.11)
  Sample $\mathbf{W}$ using equation (5.10)
  Sample $\sigma^2$ using equation (5.12)
  Sample $\alpha$ using equation (5.15)
```

5.6.2 Adaptive PPCA

In many common PPCA applications, constraints on the latent feature dimensionality occur naturally. In data visualisation, we are mostly interested in reducing high-dimensional data down to two or three dimensions; in regression problems when PCA is used to remove *multicollinearity* from input features, the output dimensionality is usually fixed to D (the dimensionality of the input). In these scenarios a multivariate hypergeometric prior on latent variables $\{\mathbf{z}_n\}_{n=1}^N$ would allow for explicit control over the number of latent subspaces; we call this the *adaptive* PPCA (aPPCA) [4]. K denotes the number of unique orthogonal linear subspaces which we will use to reduce the original data into the lower dimensional space; each input data point can be associated with a different subset of L subspaces, selected from a total of K subspaces. So, any single point is actually represented by lower dimensional spaces subsets of $\mathbb{R}^L$. Note that the orthogonality assumption $\mathbf{w}_i \perp \mathbf{w}_j \ \forall i \neq j$ for the columns of **W** implies that $K \leq D$.

The multivariate hypergeometric prior allows updates of $\mathbf{z}_{1,\ldots,N}$ across N in parallel, since the number of observed data points assigned to a latent subspace no longer implies higher probability of assigning a new data point to that subspace, i.e., no reinforcement effect. Instead, for each $n = \{1, \ldots, N\}$, we learn $\mathbf{z}_n$ by sampling the L indices $\{l_1, \ldots, l_L\}$ from categorical distribution; this is highlighted in Algorithm 6.

Algorithm 6 Update latent variables $\{\mathbf{z}_n\}_{n=1}^N$

```
for $n \leftarrow 1$ to $N$
    Set $\mathbf{z}_n = \mathbf{0}$
    for $i \leftarrow 1$ to $L$
        Sample $l_i \sim$ Categorical $\left( \frac{(1-z_{1n}) \exp\left(\left(\mathbf{y}_n^T \mathbf{w}_1\right)^2\right)}{\sum_k (1-z_{kn}) \exp\left(\left(\mathbf{y}_n^T \mathbf{w}_1\right)^2\right)}, \ldots, \frac{(1-z_{Kn}) \exp\left(\left(\mathbf{y}_n^T \mathbf{w}_K\right)^2\right)}{\sum_k (1-z_{kn}) \exp\left(\left(\mathbf{y}_n^T \mathbf{w}_K\right)^2\right)} \right)$
        Set $z_{l_i n} = 1$
```

In dimensionality reduction applications we often assume L being two or three, hence l_1 might indicate the x-axis, l_2 the y-axis and l_3 the z-axis of the lower-dimensional subspace. A Gibbs sampler for the aPPCA is suggested in Algorithm 7.

Algorithm 7 Pseudocode for inference in parametric aPPCA using Gibbs sampling.

```
Input: $\mathbf{Y}, \boldsymbol{\Theta}$, MaxIter
Initialise: Sample a random $(K \times N)$ binary matrix $\mathbf{Z}$ and initialise $\mathbf{W}$ using PCA
  for $n \leftarrow 1$ to $N$
    for $k \leftarrow 1$ to $K$
      if $z_{kn} = 1$
        Sample $x_{kn}$ using equation (5.11)
  Sample $\mathbf{z}_{1,\ldots,N}$ using Algorithm 6
  Sample $\mathbf{W}$ using equation (5.10)
  Sample $\sigma^2$ using equation (5.12)
```

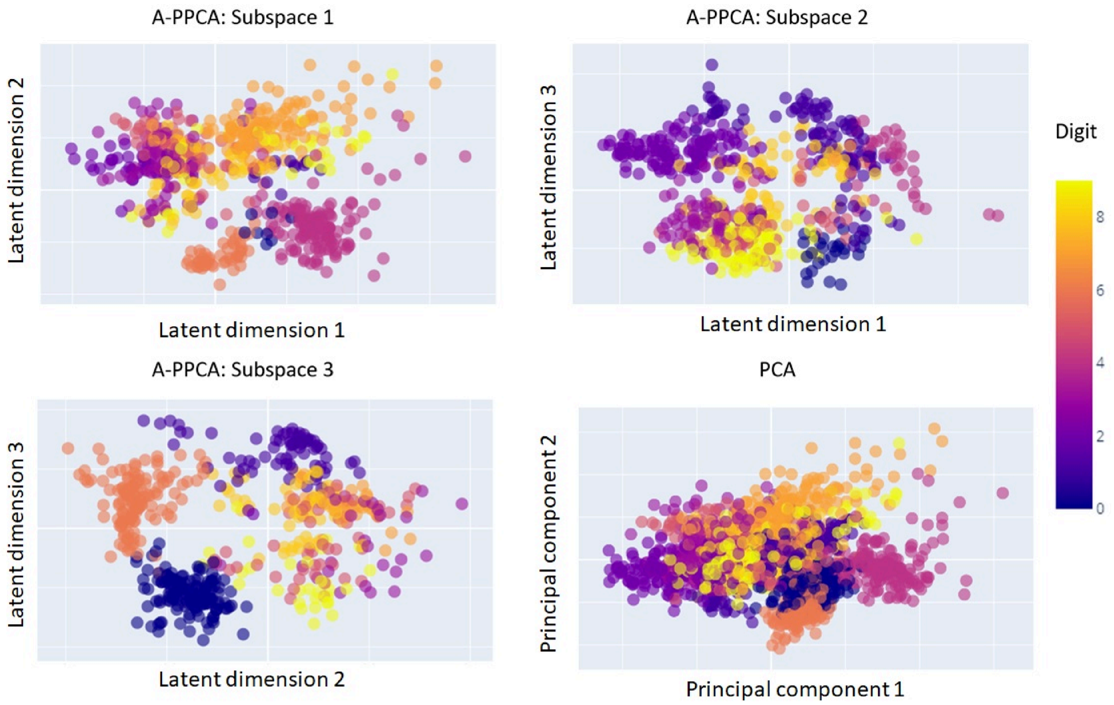


Figure 5.4: Scatter plot of the 2-dimensional projections of $10,000$ MNIST digits, obtained using aPPCA and PCA. The first three subplots contain only proportions of the data which have been estimated by aPPCA to lie in the corresponding subspace (i.e. Subspace 1 is spanned by features 1 and 2; Subspace 2 by features 2 and 3; Subspace 3 by 1 and 3). The fourth subplot shows the 2-dimensional projection of all digits obtained using PCA.

5.6.3 Experiments

Visualisation with aPPCA

Despite the increased popularity of nonlinear manifold embedding algorithms for data visualisation, linear dimensionality reduction methods remain of fundamental importance to exploratory data visualisation, arguably due to their scalability, stability, and intuitive data representation. In this section, we provide simple illustrations of how aPPCA complements conventional PCA visualisations.

Typically, we use PCA to project all the data down to the first two principal components, in aPPCA each point is also reduced to two components ($L = 2$), but these components can be computed only based on some of the data, having some larger K unique sparse principal components in total. This essentially means we visualise the data using multiple scatter plots including different subsets of the projected data, instead of the single crowded plot in PCA.

MNIST data set First, we look at subspace sharing of MNIST digits. Note that with PCA we project each data point down onto the same two orthogonal principal components preserving most variance and we display the projections in a single 2-dimensional plot. With aPPCA we can still project each point onto $L = 2$ orthogonal principal components, but the components are not all constrained to be shared for all the data if $K > L$. For more intuitive visualisation, we first use a 2-layer multilayer perceptron variational autoencoder (VAE) [112] to reduce the dimension of 10,000 MNIST digits. The 784-dimensional data is reduced with the VAE to 10 dimensions and then we train parametric aPPCA with $K = 3$ and $L = 2$ to visualise the digits in the latent space. We will assume that subspace 1 is spanned by the inferred features[2] 1 and 2; subspace 2 by features 2 and 3; subspace 3 by features 1 and 3. Note that all pairs of subspaces share one of their principal axes. In Figure 5.4 we display the reduced data in each of these subspaces where we can see an increased separation between many of the distinct clusters of different digits. From Figure 5.5 we can see that distinct geometric properties of digits are encoded in the identified subspaces. Figure 5.5 shows randomly selected digits from each subspace and we can see that most digits in subspace 1 are written in the thicker font; most digits in subspace 3 are slanted. The visualisation also reduces the crowding effect of PCA and produces multiple 2-dimensional plots which jointly decompose the data and intuitively organise the observed data.

COIL-20 data set We consider another data visualisation example, this time using data from the Columbia University Image Library (COIL-20) [113]. The data set contains low-resolution images (32×32 pixels) of 20 different objects. The objects are placed on a motorised turntable against a blank background and the turntable is rotated 360 degrees to vary object pose with respect to a fixed camera. 72 images of each object are taken, at pose intervals of 5 degrees rotation and the images are size normalised. This means that objects which are very similar at different view angles will result in very similar 72-image observations.

First, we reduce all the 1440 images onto the two principal components which are

[2] Features is a general term, however in this context each feature is a principal component.

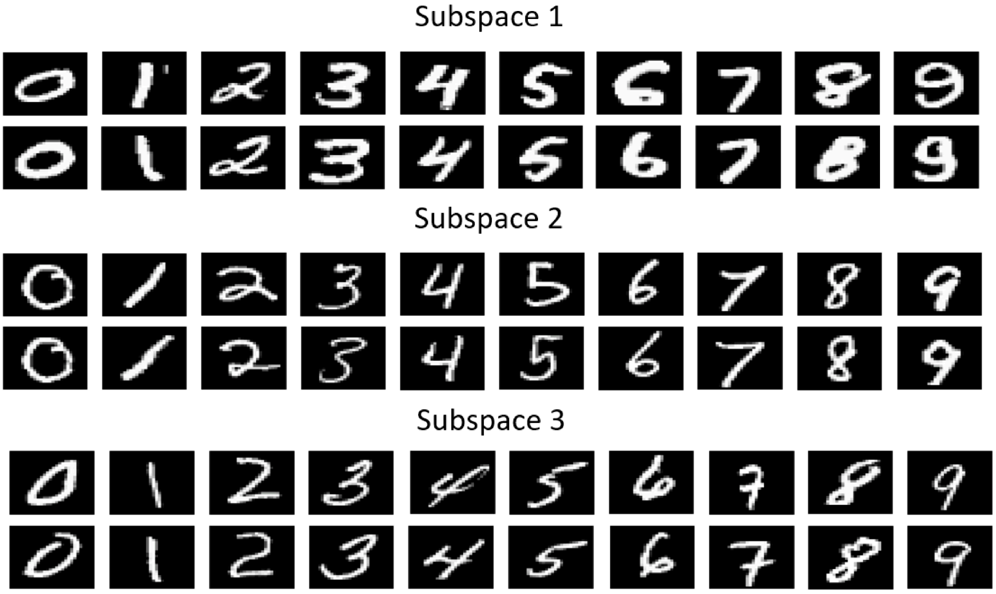
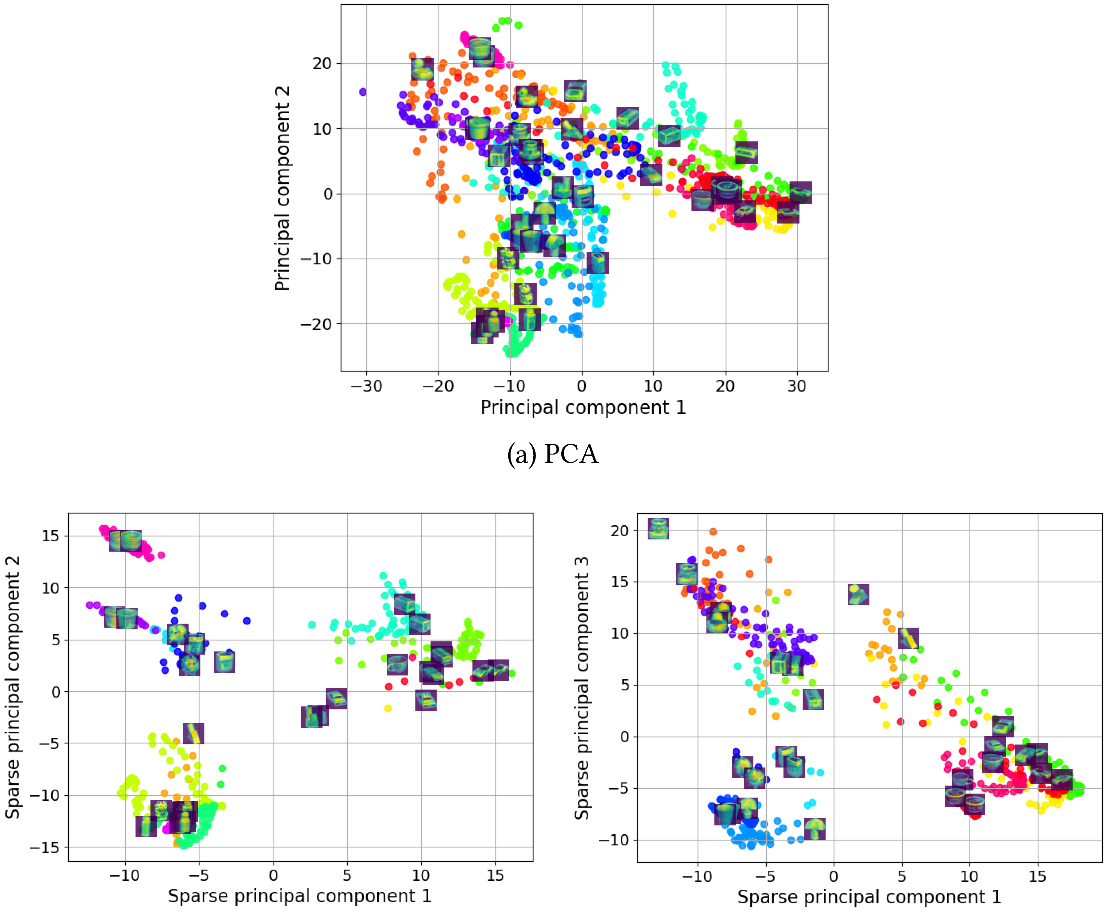


Figure 5.5: Randomly selected MNIST digits from each of the identified subspaces. The top panel consists of mostly thicker digits; the bottom panel is dominated by slanted digits.

computed to preserve the variance globally across all data points. Images from the different objects are displayed in different colours in Figure 5.6, whereas a fraction of the actual images is overlaid on the scatter plot. We see that some of the objects, such as two of the toy cars framed from a front view angle (i.e. green and yellow class on the far right of the plot in Figure 5.6 (a)), are well separated with other rectangular objects with similar geometry. However, most of the objects are bundled in the center of the plot and not recognisable in the reduced two-dimensional space.

Next, we fit an aPPCA model with $K = 4$ and $L = 2$ which effectively learns four sparse principal components with each data point associated with a subset of exactly two of these components. In (b)-(c) of Figure 5.6, we display the points sharing combinations of the estimated subspaces spanned by the sparse principal components (i.e. four unique components lead to $(4 \times 3)/2 = 6$ subspaces with some shared axes). We see that projections onto the sparse principal components reduce the crowding effect of PCA. In addition, the different sparse principal components encode interpretable geometric properties of the objects observed. For example, objects with smaller values along with the sparse principal component number two, tend to be narrower, whereas, objects with large values along the sparse principal component 1 tend to be less cylindrical. Within



(a) PCA

(b) aPPCA: subspace $\{1, 2\}$

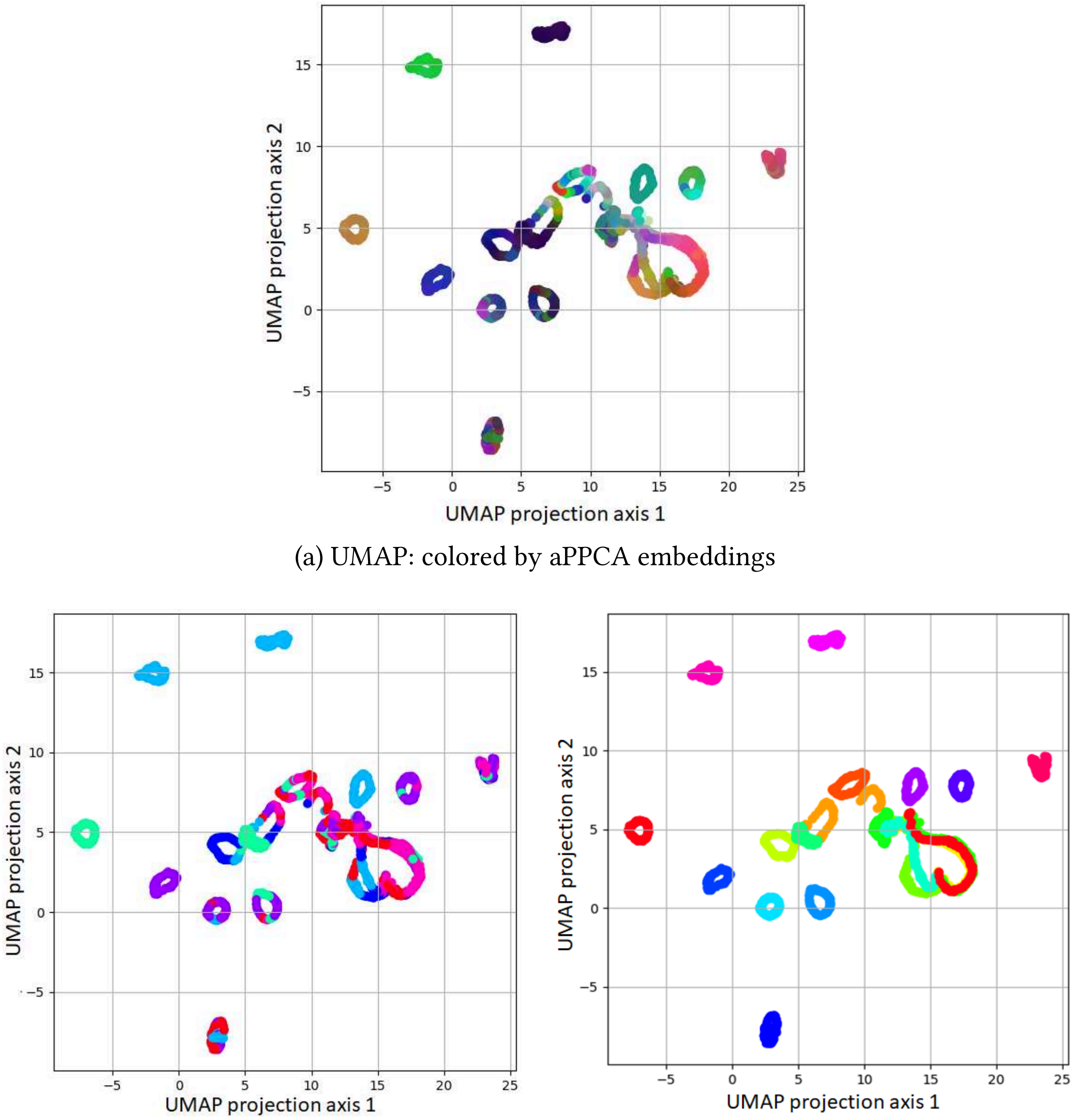
(c) aPPCA: subspace $\{1, 3\}$

Figure 5.6: 2-dimensional projections of the COIL-20 data set images using PCA and aPPCA methods. (a) The 2-dimensional scatter plot is obtained by reducing the 1024-dimensional images to 2-dimensional with PCA, and a sample of the original images is placed over their projection. (b)-(c) The two-dimensional projections of data points onto sparse principal components they are associated with, inferred using aPPCA. Where (a) includes all data points in a single projection, aPPCA in (b)-(c) identifies subsets of the data sets sharing principal components, hence principal components are estimated using only a subset of the observations (i.e., sparse principal components).

each 2-dimensional subspace, the different object projections are easier to separate and different objects with similar projections also have intuitive image similarity under a rotation angle.

Interpreting global structure in manifold embedding Toy problems such as COIL-20 have been used to showcase manifold embedding methods such as t-SNE [2] and more recently UMAP [81]. Empirically, both t-SNE and UMAP often lead to very good class separability in the lower dimensional projections, particularly in scenarios where class separability in the original high-dimensional data is good (i.e., such as for COIL-20). At the same time, it is well known that many manifold embedding algorithms such as UMAP and t-SNE do not preserve the global structure of the data manifold, unlike linear methods such as PCA and multidimensional scaling, or kernel space models such as Gaussian process latent variable models. This often leads to lower-dimensional projections which reflect class separability well when captured in localised regions of the manifold (such as in MNIST and COIL-20) but do not capture similarities across different classes adequately. To illustrate, Figure 5.7 shows the two-dimensional projections of COIL-20, obtained using UMAP. Figure 5.7c) shows objects from the same class are associated with the same colour. Certain objects have been separated into 2 or 3 clusters (i.e., the duck and the bowl), depending on the angle of view, but if the aim is object classification based on the two-dimensional embedding of the data, the task is nearly trivial. The challenge is less clear if we are looking to uncover latent structures between the objects.

[81] has suggested using PCA to reduce data onto its first three PCs and colour UMAP embeddings using RGB values defined by the 3D PCA projections of each point. This approach suggests that points close in the PCA projection of the data would also have a similar colour. By contrast, as colours transition, this means that data points are projected far apart on some of the PCs. The problem with using PCA as diagnostics for UMAP projections in this manner is that we are likely to crowd observations, overestimating proximity between most points due to the simplistic, global assumptions of PCA. If we are interested in using manifold embedding methods such as UMAP, which



(a) UMAP: colored by aPPCA embeddings

(b) UMAP: colored by shared subspaces

(c) UMAP: colored by classes

Figure 5.7: 2D projections of the COIL-20 data set images using UMAP. The x-axis and y-axis are determined based on the UMAP projection. In (a) the colours encode the non-zero 3D projection of the points performed using aPPCA. Each point is associated with exactly three sparse PCs, but the total number of components is larger. In (b) the colours encode subspace sharing with the same colours (i.e., points in the same colour share at least two sparse PCs). In (c) the colours encoded indicate the object classes.

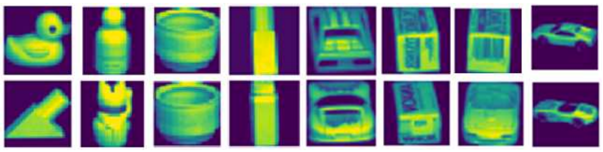


Figure 5.8: Example images from different object classes in the COIL-20 data set, with shared subspaces and proximity in the three-dimensional orthogonal aPPCA projection of input images. Proximity was defined with basic K-means clustering of the lower dimensional projections, where Figure 5.6 shows how clustered specific subspaces are. Note that objects sharing subspaces are merely estimated to share a covariance structure.

preserve the local structure of the original manifold, we can instead use piecewise linear methods such as aPPCA, which capture the global structure of the manifold, and use these to annotate the 2D UMAP projections (Figure 5.7b). In this figure, we use different symbols to denote points associated with different subspaces; the colours depend on the 3D projection obtained with a single run of aPPCA with $K = 4$ and $L = 3$ (i.e. leading to four subspaces spanned by sparse PCs $\{1, 2, 3\}$; $\{2, 3, 4\}$; $\{1, 2, 4\}$ and $\{1, 3, 4\}$). Note that under this diagnostic, similar colours (in RGB values) indicate similarity in the reduced form. We can see that aPPCA much of the omitted cross-object similarities is specific to certain rotations: the rotated Maneki-neko (i.e., lucky cat figurine) and cylindrical bottle; the duck toy and the similar shape wooden part; the different clusters of bowl images and others. To further aid intuition, we have also included images of rotated object similarities identified using subspace decomposition diagnostics with aPPCA, see Figure 5.8.

Data pre-processing

Another ubiquitous use of PCA is *data whitening*. This is an often-used pre-processing step that aims to *decorrelate* the observed data to simplify subsequent processing and analysis, for example, image data tends to have highly correlated adjacent pixels. In this capacity, PCA works by 'rotating' the data in observation space, retaining dimensionality unlike with visualisation applications.

Here we show a simple example demonstrating how aPPCA can be used to do more effective *local* whitening which can lead to more accurate and interpretable supervised

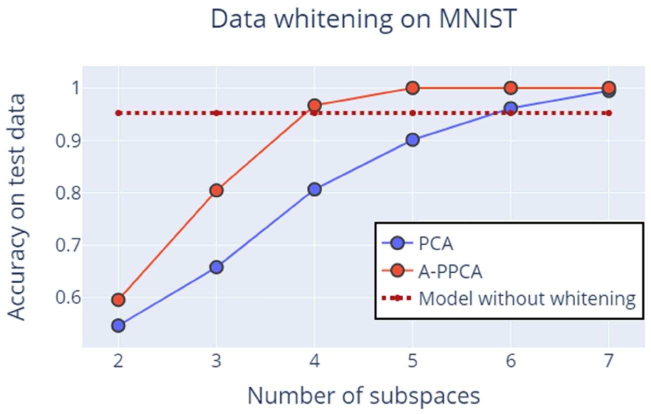


Figure 5.9: Classification accuracy of a multilayer perceptron evaluated using 10,000 MNIST digits in three different setups: no whitening; data pre-processed with PCA; data pre-processed with aPPCA. On the x-axis we show the number of reduced dimensions for different instances of the same classifier. The y-axis indicates the out-of-sample accuracy, evaluated using 10-fold cross-validation.

classification in decorrelated latent feature space. To demonstrate this, we compare a classifier trained on raw data with the same classifier trained on the first few principal components projections of the data where the components are estimated (1) globally using PCA and (2) locally, within subsets of the data using aPPCA.

For simplicity, we show an example of pre-processing the MNIST handwritten digit classification data set, before training a multilayer perceptron. We train a simple multilayer perceptron with one hidden layer with a softmax activation function evaluated using 10-fold cross-validation on $10,000$-image subset of the 784-dimensional MNIST data set. We compare the performance of the same classifier network when: (1) Trained on the original 784-dimensional pre-processed data, (2) Trained on the lower-dimensional projection of the data using PCA, and (3) Trained on data locally whitened by aPPCA (K-dimensional). The classifier is a multilayer perceptron in all three scenarios. Figure 5.9 shows the classification accuracy of these three different pre-processing approaches as we vary K, i.e., the number of the principal component onto which we can project the data down. For aPPCA, we have kept $L = K - 1$ for simplicity. Intuitively, we also see increases in performance if multiple, separate classifiers are trained on each L-dimensional subspace, but usually, after whitening with PCA, a single classifier is used.

A key feature of the aPPCA for localised data whitening is that it estimates more robust subspaces which can be seen in the smaller number of subspaces (i.e., principal components or columns of **W**) required for training of the same classifier, to achieve better performance. The multilayer perceptron trained on PCA whitened data requires more subspaces in training to achieve comparable performance.

Blind source separation in fMRI

Functional magnetic resonance imaging (fMRI) is a technique for the non-invasive study of brain function. fMRI can act as an indirect measure of neuronal activation in the brain, by detecting *blood oxygenation level dependent* (BOLD) contrast [114]. BOLD relies on the fact that oxygenated (diamagnetic) and deoxygenated (paramagnetic) blood have different magnetic properties. When neurons fire there is a resultant increase in localised flow of more oxygenated blood, which can be detected using BOLD fMRI.

fMRI time-series data is often represented as a series of three-dimensional images (see Figure 5.10). However, data can be also represented as a two-dimensional matrix using vectorised voxel matrices over time (time by voxels). In this representation, each matrix row contains all voxels from the brain image (or the subset selected for analysis) from a single time instance. Although useful, fMRI data often suffers from a low image contrast-to-noise ratio, it is biased by subject head motions, scanner drift (i.e., due to equipment overheating), and signals from irrelevant physiological sources (cardiac or pulmonary). Therefore, direct analysis of raw fMRI measurements is rare [115] and domain experts tend to work with pre-processed, reduced statistics of the data. In clinical studies, due to the typical scarcity of fMRI series per subject and the low signal-to-noise ratio, flexible black-box algorithms are rarely used. The preferred methods for pre-processing of fMRI series and localisation of active spatial regions of the brain are variants of linear dimensionality reduction methods such as PCA and FA [115–119]. Typically, of primary interest is the analysis of a representative subset of the inferred principal components or factors respectively, instead of the use of raw data.

A key problem with this approach is that these linear methods assume that the components/factors are a linear combination of all the data, i.e., in other words, PCA and FA assume that all components are *active* for the full duration of the recording. Common implementations for fMRI series [120, 121] might adopt thresholding the inferred components or using sparse versions of the decomposition techniques. These can still lead to biased decomposition into components, and we are likely to overestimate the firing area of the brain for some components and completely overlook functional areas of the brain which are active for short periods of time. Here, we show that our proposed adaptive linear methods, are better-motivated models for alleviating this problem and can infer better localised spatial regions of activation from fMRI. Furthermore, we can potentially discover novel short-term components in a principled, probabilistic, data-driven fashion.

As a proof of concept, here we apply aPPCA to fMRI data collected from a single participant while exposed to continuous visual stimuli. fMRI data were initially realigned to correct for subject motion and registered to a group template (Montreal Neurological Institute Template). Using a 3T Siemens scanner, a whole-brain image with a voxel resolution of $(2 \times 2 \times 2)$ mm was acquired every 0.8 seconds. The data had $215,302$ voxels and 989 time instances. aPPCA decomposition was performed by treating time instances as features, which is a standard procedure in the neuroimaging field. For aPPCA we used $K = 500$ unique components and constraint of $L = 200$ components, which were selected to achieve component similarity with the benchmark and enable visually intuitive comparisons. We also performed PPCA with $K = 200$ components for comparison, see Figure 5.11.

The figure shows the component most associated with the task estimated both with aPPCA and PPCA. aPPCA results in sparser maps across space, which enhance localisation. This sparsity increases with higher numbers of components that explain less variance in the data. This can be useful for identifying noisy components and brain areas that are only transiently active during task performance. We also show the corrected t-statistic map (Figure 5.11) which shows the voxels that have a significant correlation with the visual stimuli. The map is family-wise error (FWE) rate corrected at $p < 0.05$

at voxel threshold $p < 0.001$. One benefit of decomposition methods versus standard correlation methods is that they do not need a predefined model of assumed task activation.

Direct quantitative evaluation of pre-processing tools for fMRI data is an open problem, due to the lack of a clear ground-truth definition of brain-activity-related components. We have measured the mean reconstruction error across all $215,302$ voxels as well as the standard deviation across voxels. We find that the highest error with the highest standard deviation (i.e. average root mean square error (RMSE) of **16.5**, the standard deviation of RMSE of **4.8**) was obtained using PPCA. aPPCA reconstruction gradually reduces these errors depending upon the ratio of K and L used, with the best scoring reconstruction having an average RMSE of **14.1** and standard deviation RMSE (across voxels) of **3.0**. The lower standard deviation of error across voxels supports our hypothesis of better-preserved local region information using aPPCA. Due to the simplicity of the imaging setup, both methods were able to identify components highly correlated to the stimuli, see Figure 5.11. The typical goal for experts would be to examine functions of the specific brain regions or networks, as well as potentially affected areas of the brain after head trauma or stroke.

The common analysis practice would be to threshold the observation-specific loadings (i.e., reduced form data) and only consider voxels that *significantly* contribute to selected subsets of components. The adaptive nature of aPPCA allows us to infer the voxels association with specific components (i.e., $\mathbf{Z}$ switches off voxels not part of a component) in a principled fashion as a part of a fully probabilistic model. In addition, the experimental user has explicit control over the contrast voxels used in different components (ratio of K and L) and this can be useful for achieving better spatial localisation, without thresholding which is an inherently subjective procedure.

5.7 Discussion

In this chapter, we have studied generic discrete latent variable augmentation for ubiquitous linear Gaussian methods applied for feature learning, whitening, and dimensionality reduction applications. The shortcomings of existing linear Gaussian methods are showcased, and the alternative models are derived using the multivariate hypergeometric distribution. This creates two novel models, the aFA and the aPPCA model which can be trained efficiently yet overcome the inherent over-partitioning in beta processes and allows for more flexible regularisation of the model capacity, compared to Beta-Bernoulli models. The proposed models can be extended to many other related methods such as generalised linear Gaussian models, Gaussian process latent variable models (GPLVMs), kernel PCA methods, and others. [122] has already introduced the problem of handling discontinuity in GPLVMs and proposed a simple *spike and slab prior* to augment the continuous latent variables in GPLVMs. Augmenting GPLVMs with discrete multivariate hypergeometric feature allocation indicators, would in principle, allow for a richer and more compact model of the manifold using a smaller number of underlying, feature-specific Gaussian processes.

In the experiments, we show that compared to the other variants of FA, the aFA gives the lowest reconstruction error. Furthermore, the aFA was applied to nearly raw digits to show that images of visually similar digits share more factors than visually distinct digits. For the aPPCA model, we have also proposed efficient practical inference methods for distributions on Stiefel manifolds. The utility of the proposed tools is demonstrated on a wide range of synthetic latent feature Gaussian data sets, MNIST handwritten digit images, COIL-20 object images, and brain imaging fMRI data. The synthetic data study shows that a wide range of feature allocation distributions can be captured with a multivariate hypergeometric model. We have applied aPPCA to MNIST variational autoencoder projections, to show that it can be used to identify images sharing clear geometric features. We conclude with an application of aPPCA to a widely encountered problem in brain imaging with fMRI and demonstrate an accurate decomposition of active spatial regions in the brain during different stimuli (or at rest). We also demonstrate that this discrete-continuous decomposition leads to more accurate localisation of active brain regions. This finding has the potential to lead to significant improvements in analysis pipelines for fMRI data for neurological screening and cognitive neuroscience applications.

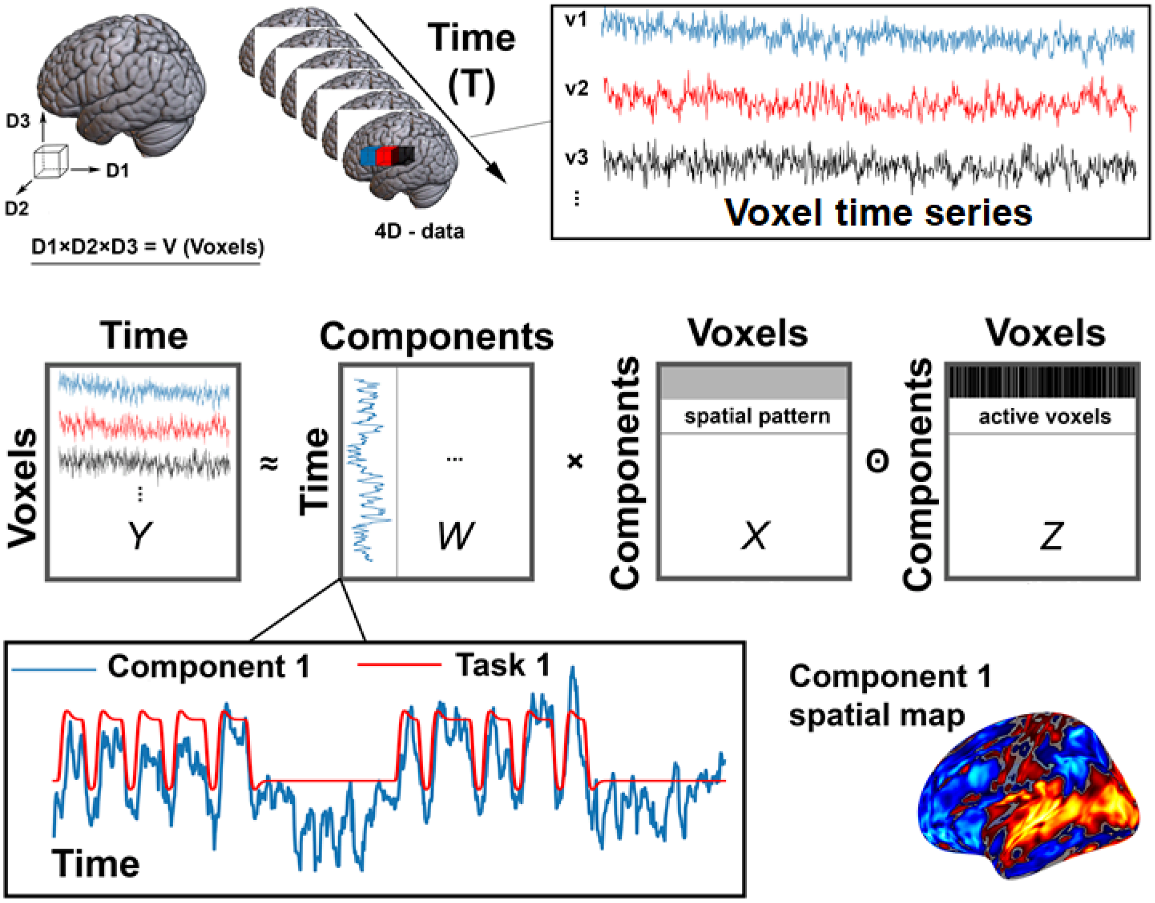


Figure 5.10: fMRI data of 3-dimensional brain volumes were collected over time (e.g., every 0.8 seconds). Typically, images are vectorised and represented as two-dimensional **T** × **V** matrices (top panel), with **V** being a number of all voxels in all dimensions and **T** the number of time instances. This matrix can be then reduced down to a **K** × **V** matrix (i.e., **X**) which represents spatial maps of regions with intrinsically similar time-courses (middle panel). **W** denotes the modelled transformation matrix and **Z** indicates whether components (i.e., rows of **X**) should be included in the representation of the data matrix or not. Columns of **W**, also referred to as components, are easier to interpret in terms of their correlation to experimental stimuli.

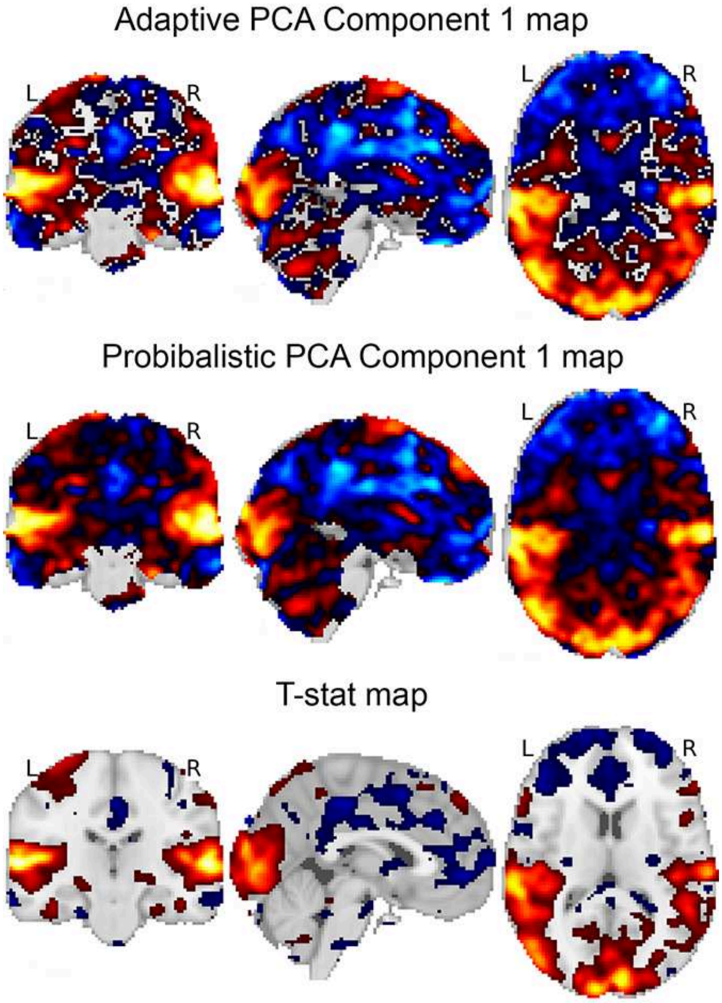

Adaptive PCA Component 1 map
L
R
L
R
Probibalistic PCA Component 1 map
L
R
L
R
T-stat map
L
R
L
R

Figure 5.11: Lower dimensional fMRI recording reduced across time, plotted against the subject brain. The fMRI time series of length T is reduced to K components and here we display the single component most associated with the stimuli during the experiment. The top panel displays the reduced projection estimated using aPPCA and the middle panel is the projection estimated using PPCA. The larger number of grey regions indicates that aPPCA projection better localises the regions of the brain fluctuation through time, as a response to the visual stimuli. Reference regions of activation can be seen from the *t-map* in the bottom panel displaying the correlation of the component with the ground-truth visual stimuli.

Chapter 6

Bayesian nonparametric extensions

The processes discussed in Section 2.2 have been used extensively as priors on many different applications such as (not limited to) clustering [55, 123], dimensionality reduction [85, 106, 124], hidden Markov models [125], and more recently model misspecification [78]. However, all these applications restrict the latent space to be binary, for example in clustering an observation can either belong to a cluster or not belong to a cluster. This is often too restrictive, as one may want the latent space to include additional information such as the level of membership to a cluster; this can be done if the latent space is no longer binary.

[19] proposed the beta-negative-binomial process to represent the latent space to be count in modelling document data, however, sampling from such a process requires inefficient sampling techniques such as stick-breaking. Therefore, this chapter builds upon the work proposed in [6, 7] by introducing two novel marginal processes which generalise the Indian buffet processes from Section 2.2. We first introduce two Poisson point processes, the first process is called the *discrete marked beta-binomial process*, and the second process is called the *marked beta-negative-binomial process*. We then derive a marginal process that allows for efficient sampling for the posterior, we call these the *multi-scoop Indian buffet process* and the *infinite-scoop Indian buffet process*; the name is motivated by the Indian buffet process metaphor used to describe the beta-Bernoulli process; which is a special case of the proposed processes.

6.1 Discrete marked beta-binomial process

6.1.1 Discrete marked beta process

The *discrete marked beta process* (dmBP) is similar to the beta process discussed in section 2.2.2, however a set-of-three of points $\{(\theta_k, \pi_k, m_k)\}_{k=1}^{\infty}$ are drawn form it; such that $(\theta_k, \pi_k, m_k) \in \mathbf{\Omega} \times [0,1] \times \mathbb{N}_0$ for all k. These points are drawn from Poisson process with the Lévy measure:

$$\nu\left(d\theta, d\pi, dm\right) = \alpha\left(\theta\right) \pi^{-1} \left(1-\pi\right)^{\alpha(\theta)-1} d\pi H_0\left(d\theta\right) M_0\left(dm\right),$$

where $\alpha\left(\cdot\right)$ is a concentration function, H_0 is a continuous measure and M_0 is a discrete measure. Draws from the dmBP are:

$$MH|\alpha_0.M_0H_0 \sim \text{dmBP}\left(\alpha_0, M_0H_0\right),$$

$$MH = \sum_k \pi_k \delta_{(m_k,\theta_k)},$$

where the concentration function $\alpha\left(\cdot\right) = \alpha_0 \in \mathbb{R}^+$ is now a concentration parameter.

6.1.2 Binomial process

Points from the *Binomial process* (BinP) are drawn from Poisson process with the Lévy measure:

$$\lambda\left(d\theta, d\pi, dm\right) = \sum_{i=1}^{m} \delta_i\left(d\pi\right) H\left(d\theta\right) dm.$$

Note that the above is equivalent to sampling from m Bernoulli processes (see Section 2.2.2). Draws from the binomial process can be represented as:

$$B_n|MH \sim \text{BinP}\left(MH\right),$$

$$B_n = \sum_k b_{nk} \delta_{(m_k,\theta_k)},$$

$$b_{nk}|m_k, \pi_k \sim \text{binomial}\left(m_k, \pi_k\right),$$

where points $(\theta_k, b_{nk}) \in \mathbf{\Omega} \times \{0, 1, 2, \ldots, m_k\}$.

6.1.3 Conjugacy

The discrete marked beta process is conjugate to the binomial process. Let $MH|\alpha_0, M_0H_0 \sim$ dmBP$\left(\alpha_0, M_0H_0\right)$ and let $\{B_n\}_{n=1}^{N}|MH \overset{i.i.d.}{\sim}$ BinP$\left(MH\right)$ be N independent samples from the binomial process (with hazard measure MH), then the posterior update on MH is still a discrete marked beta process:

$$MH|M_NH_N, \alpha_N\left(\theta\right)\{B_n\}_{n=1}^{N} \sim \text{dmBP}\left(\alpha_N\left(\theta\right), M_NH_N\right),$$

where:

$$\begin{aligned} M_NH_N =& \frac{\alpha_0}{\alpha\left(\theta\right)}M_0H_0 + \sum_k \frac{n_k}{\alpha\left(\theta\right)}\delta_{(m_k,\theta_k)}, \\ \alpha_N\left(\theta\right) =& \alpha_0 + Nm_k, \end{aligned}$$

where $n_k = \sum_n b_{nk}$ is the number of times in which the atom at location θ_k appears in $\{B_n\}_{n=1}^{N}$.

It is useful to marginalise over the MH and work directly with the posterior marginal process. However, this is impossible as this marginal is not available in closed form. Therefore, for the remainder of the section, we'll focus on a special case of this process, which is the beta-binomial process.

6.1.4 Special case: beta-binomial process

The beta-binomial process is a special case of the discrete marked beta-Binomial process from Section 6.1, where the discrete marked beta process is replaced with the beta process.

Beta process A set-of-two of points $\{(\theta_k, \pi_k)\}_{k=1}^{\infty}$ are drawn from beta process; such that $(\theta_k, \pi_k) \in \mathbf{\Omega} \times [0, 1]$ for all k. These points are drawn from Poisson process with the Lévy measure:

$$\nu\left(d\theta,d\pi\right)=\alpha\left(\theta\right)\pi^{-1}\left(1-\pi\right)^{\alpha(\theta)-1}d\pi H_0\left(d\theta\right),$$

where $\alpha\left(\cdot\right)$ is a concentration function, H_0 is a continuous measure. Draws from the BP are:

$$\begin{aligned}H|\alpha_0,H_0&\sim\text{BP}\left(\alpha_0,H_0\right),\\ H&=\sum_k\pi_k\delta_{\theta_k},\end{aligned}$$

where the concentration function $\alpha\left(\cdot\right)=\alpha_0\in\mathbb{R}^+$ is now a concentration parameter.

Binomial process Draws from the binomial process can be represented as:

$$\begin{aligned}B_n|mH&\sim\text{BinP}\left(mH\right),\\ B_n&=\sum_k b_{nk}\delta_{\theta_k},\\ b_{nk}|m,\pi_k&\sim\text{binomial}\left(m,\pi_k\right),\end{aligned}$$

where points $\left(\theta_k,b_{nk}\right)\in\mathbf{\Omega}\times\{0,1,2,\ldots,m\}$ and $m\in\mathbb{N}_0$.

Conjugacy The beta process is conjugate to the binomial process, let $H|a_0,H_0\sim$ BP $\left(\alpha_0,H_0\right)$ and let $\{B_n\}_{n=1}^N|mH\overset{i.i.d.}{\sim}$ BinP $\left(mH\right)$ be N independent samples from the binomial process (with hazard measure mH), then the posterior update on H is still a beta process:

$$H|\{B_n\}_{n=1}^N,\alpha_N,H_N\sim\text{BP}\left(\alpha_N,H_N\right),$$

where:

$$\begin{aligned}H_N=&\frac{\alpha_0}{\alpha_N}H_0+\sum_k\frac{n_k}{\alpha_N}\delta_{(\theta_k)},\\ \alpha_N=&\alpha_0+Nm,\end{aligned}$$

where $n_k = \sum_n b_{nk}$ is the number of times in which the atom at location θ_k appears in $\{B_n\}_{n=1}^N$.

6.1.5 Distribution on infinite count matrices

In this section we'll derive the probability distribution over an infinite count matrix; this would generalise the work done in [8].

Finite Let $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]$ be a K-dimensional vector; where $\pi_k \in [0, 1]\ \forall k$. The prior distribution over each π_k is the beta distribution:

$$\pi_k | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right),$$

where $\alpha > 0$ is a hyperparameter. Let **Z** be a $(N \times K)$ where each element $z_{nk} \in \{0, 1, \dots, m\}$ is independently and identically sampled from a binomial distribution:

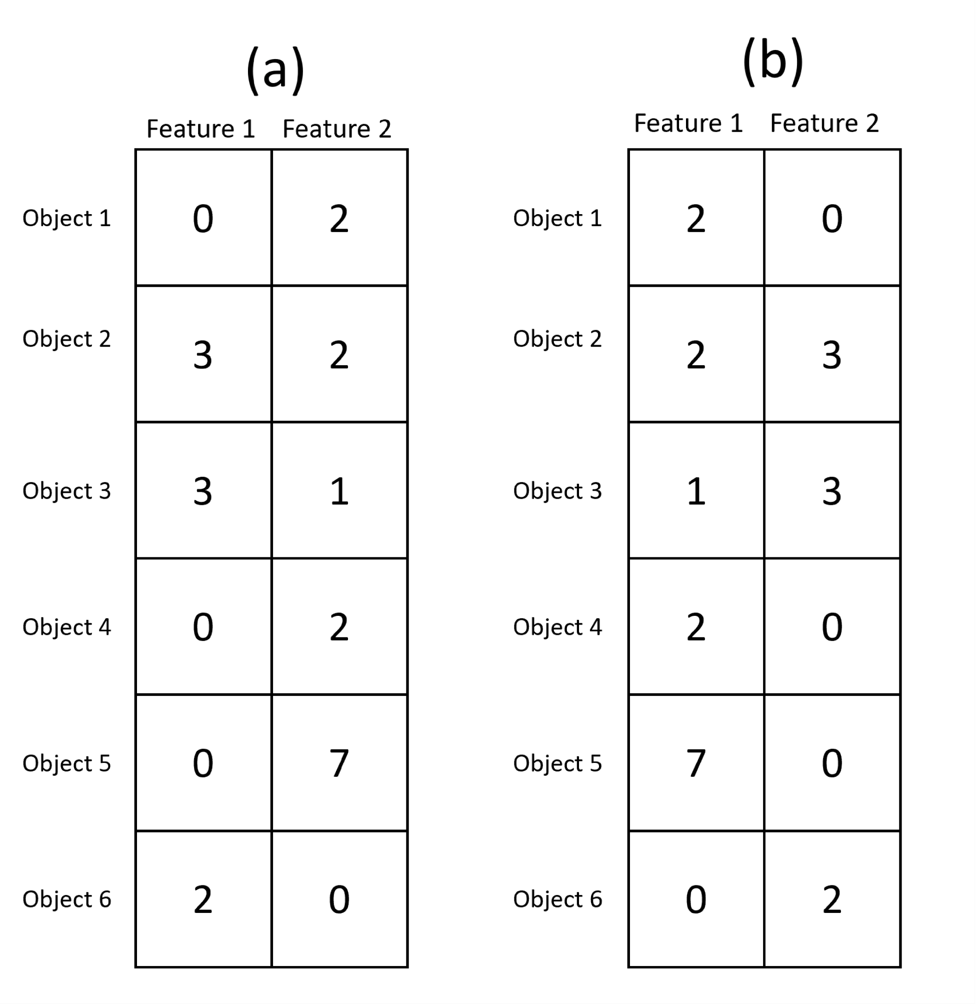
$$z_{nk} \overset{i.i.d.}{\sim} \text{binomial}\left(m, \pi_k\right),$$

where $n \in \{1, \dots, N\}$, $k \in \{1, \dots, K\}$, and $m \in \mathbb{N}_0$ is the 'number of trails' parameter of the binomial distribution. From a latent feature perspective z_{nk} can be interpreted as the number of times an object n possessing feature k. The marginal likelihood of **Z** is:

$$\begin{aligned}
&\mathrm{P}\left(\mathbf{Z}|, \alpha, m\right) \\
&= \prod_{k=1}^{K} \int \left(\prod_{n=1}^{N} \mathrm{P}\left(z_{nk}|\pi_k, m\right) \right) \mathrm{P}\left(\pi_k|m\right) d\pi_k \\
&= \prod_{k=1}^{D} \left(\prod_{n} \frac{(m!)}{(m - z_{nk})!\,(z_{nk})!} \right) \left(\frac{\alpha}{K}\right) \frac{\Gamma\left(n_k + \frac{\alpha}{K}\right) \Gamma\left(Nm - n_k + 1\right)}{\Gamma\left(\frac{\alpha}{K} + Nr + 1\right)},
\end{aligned} \tag{6.1}$$

where $n_k = \sum_{n=1}^{N} z_{nk}$; see Appendix D.1 for the derivation.

Left order equivalence The marginal probability of the count matrix **Z** does not depend on the order of the features i.e., features are exchangeable. This results in having multiple matrices **Z** which encode the same feature assignments for a number of objects,



(a)

	Feature 1	Feature 2
Object 1	0	2
Object 2	3	2
Object 3	3	1
Object 4	0	2
Object 5	0	7
Object 6	2	0

(b)

	Feature 1	Feature 2
Object 1	2	0
Object 2	2	3
Object 3	1	3
Object 4	2	0
Object 5	7	0
Object 6	0	2

Figure 6.1: A plot of different count matrices. Both (a) and (b) are different matrices but they both encode the same information.

an example of this can be seen in Figure 6.1.

We extend the left-ordered form lof $(\cdot)$ from equation (2.6) to consider the same feature assignments (with counts) over different feature arrangements. This function order the columns of the matrix using the history of the columns, which ensures that the nth row is more significant than the $(n+1)$th row. Let m be the maximum value of **Z**, then the history for column k can be calculated using the following:

$$H_k = \sum_{n=1}^{N} (m+1)^{N-n} z_{nk},$$

where H_k denotes the history for column k. Then H_k can take the maximum value of $(m+1)^{N-1} + (m+1)^{N-2} + ... + (m+1)^{1} + 1 = (m+1)^{N} - 1$. Then the cardinality of $[\mathbf{Z}] = \frac{K!}{\prod_{h=0}^{(m+1)^N-1} K_h!}$ is the number of matrices that map to the same left-ordered form where K_h denotes the number of columns with history h.

Infinite count matrix By using the methods from above, the marginal probability of lof-equivalent class of binary matrices $[\mathbf{Z}]$ is:

$$\begin{aligned}
\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m\right) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} \mathrm{P}\left(\mathbf{Z}|\alpha, m\right) \\
&= \frac{K!}{\prod_{h=0}^{(m+1)^N-1} K_h!} \\
&\quad \times \prod_{k=1}^{K} \left(\prod_{n} \frac{(m!)}{(m-z_{nk})!\,(z_{nk})!} \right) \left(\frac{\alpha}{K}\right) \frac{\Gamma\left(m_d + \frac{\alpha}{K}\right) \Gamma\left(Nm - n_k + 1\right)}{\Gamma\left(\frac{\alpha}{K} + Nm + 1\right)},
\end{aligned}$$

and as $K \to \infty$, the above becomes:

$$\begin{aligned}
\lim_{K\to\infty} \left(\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m\right)\right) &= \frac{\alpha^{K^+}}{\prod_{h=1}^{(m+1)^N-1} K_h} \\
&\quad \times \exp\left(-\alpha H_{Nm}\right) \prod_{k=1}^{K^+} \frac{\prod_{n=1}^{N} \binom{m}{z_{nk}} \left(n_k - 1\right)!\,\left(Nm - n_k\right)!}{(Nm)!},
\end{aligned} \tag{6.2}$$

where $H_{Nm} = \sum_{j=1}^{Nm} \frac{1}{j}$; also known as the (Nm)th harmonic number; see Appendix D.1 for the derivation.

6.1.6 Multi scoop Indian Buffet Process

In this section we'll derive the marginal process of the beta-binomial process. We will call this the the *multi scoop Indian Buffet Process* (msIBP) as the IBP is the special case of it. To derive the process we'll continue to use a cuisine metaphor of an Indian buffet with an infinite number of dishes, where a customer can take multiple scoops of each dish; the dish refers to a feature, and the scoop refers to the number of times a feature is active. This section will be split up into three sub-sections.

New customers

When the first customer enters a buffet, they are given m 'chances' to sample a new dish (which has previously not been sampled); each chance is a sample from a Poisson distribution, the number of new dishes sampled by the first customer on the jth chance are denoted as K^{nj}_{new}, which is sampled from:

$$K^{1j}_{new} \sim \text{Poisson}\left(\frac{\alpha}{1m-(m-j)}\right),$$

where $\alpha > 0$ is some constant. More generally, the m chances for the first customer are:

$$\begin{aligned}
K^{11}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{1m-(m-1)}\right),\\
K^{12}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{1m-(m-2)}\right),\\
&\vdots\\
K^{1m}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{1m}\right).
\end{aligned}$$

The m chances for the nth customer are:

$$\begin{aligned}
K^{n1}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{nm-(m-1)}\right),\\
K^{n2}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{nm-(m-2)}\right),\\
&\vdots\\
K^{nm}_{new} &\sim \text{Poisson}\left(\frac{\alpha}{nm}\right).
\end{aligned}$$

The joint distribution of K^{ij}_{new} for $i \in \{1,\dots,N\}$ and $j \in \{1,\dots,m\}$ is:

$$\begin{aligned}
&\text{P}\Big(K^{11}_{new}, ..., K^{1m}_{new}, ..., K^{N1}_{new}, ..., K^{Nm}_{new}\Big)\\
&\qquad = \frac{\exp\left(-\frac{\alpha}{1}\right)\left(\frac{\alpha}{1}\right)^{K^{11}_{new}}}{K^{11}_{new}!} \times ... \times \frac{\exp\left(-\frac{\alpha}{Nm}\right)\left(\frac{\alpha}{Nm}\right)^{K^{Nm}_{new}}}{K^{Nm}_{new}!}.
\end{aligned}$$

The exponent of which is:

$$\begin{aligned}\exp\left(-\frac{\alpha}{1}\right) \times ,.., \times \exp\left(-\frac{\alpha}{Nm}\right) \\ = \exp\left(-\alpha\left(1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{Nm}\right)\right) = \exp\left(-\alpha H_{Nm}\right),\end{aligned}$$

and the total number of dishes sampled are $K^{+} = K^{11}_{new} + ... + K^{1m}_{new} + ... + K^{N1}_{new} + ... + K^{Nm}_{new}$, which results in:

$$\begin{aligned}&\mathrm{P}\Big(K^{11}_{new}, ..., K^{1m}_{new}, ..., K^{N1}_{new}, ..., K^{Nm}_{new}\Big) \\ &= \exp\left(-\alpha H_{Nm}\right)\left(\frac{\alpha^{K^{+}}}{\prod_{i=1}^{N}\prod_{j=1}^{m} K^{ij}_{new}!}\right) \times \left(\frac{1}{\prod_{i=1}^{N}\prod_{j=1}^{m}\left(im-(m-j)\right)^{K^{ij}_{new}}}\right).\end{aligned}$$

Existing dishes

Each customer is allowed to try a total of m number of scoops on every existing dish (i.e., a dish that has been already sampled), such that customer n will take the jth scoop from dish k with probability $\frac{n_k}{nm-(m-j)}$, where n_k is the number of scoops taken from dish k from all customers, and by construction $j \in \{0, 1, \ldots, m\}$. This information is stored in a $(N \times K^{+})$ count matrix **Z**; where $z_{nk} \in \{0, 1, \ldots, m\}$ denotes the number of scoops customer n had of dish k. Then the joint probability of the number of scoops for dish k is:

$$\mathrm{P}\left(z_{1k}, \ldots, z_{Nk}\right) = \left(\prod_{n=1}^{N}\frac{(m!)}{(m-z_{nk})!\,(z_{nk})!}\right)\frac{(n_k-1)!\,(Nm-n_k)!}{(Nm)!} \times \left(im-(m-j)\right),$$

where the left-most component of the product takes the different permutations of customer n into consideration (i.e., customer n will try z_{nk} scoops out of m total scoops for dish k), and the right-most component is there to offset the fact that customer i first sampled dish k on the jth chance. More generally, for N customers we obtain the following distribution overall K^{+} existing dishes:

$$\prod_{k=1}^{K^{+}}\left(\left(\prod_{n=1}^{N}\frac{(m!)}{(m-z_{nk})!\,(z_{nk})!}\right)\frac{(n_k-1)!\,(Nm-n_k)!}{(Nm)!}\right) \times \left(\prod_{i=1}^{N}\prod_{j=1}^{m}\left(im-(m-j)\right)^{K^{ij}_{new}}\right).$$

Combining existing and new dishes

By combining the results from the two previous sections, the probability of a particular count matrix $\mathbf{Z}$ being produced by this process is:

$$\begin{aligned}&\mathrm{P}\left(\mathbf{Z}|\alpha\right) = \\ &\exp\left(-\alpha H_{Nm}\right)\left(\frac{\alpha^{K^{+}}}{\prod_{i=1}^{N}\prod_{j=1}^{m}K_{new}^{ij}!}\right)\prod_{k=1}^{K^{+}}\left(\frac{\prod_{n=1}^{N}\binom{m}{z_{nd}}\left(n_k-1\right)!\left(Nm-n_k\right)!}{\left(Nm\right)!}\right),\end{aligned}$$

As discussed in the section above, the extended lof $(\cdot)$ can be used to determine the number of different matrices that encode the same feature assignment; see Figure 6.1. The cardinality of $[\mathbf{Z}] = \frac{\prod_{i=1}^{N}\prod_{j=1}^{m}K_{new}^{ij}!}{\prod_{h=0}^{(m+1)^N-1}K_h!}$ is the number of matrices that map to the same left-ordered form (i.e., the number of matrices that will encode the same feature assignment); where K_h denotes the number of columns with history h. Then the probability of a particular feature assignment $[\mathbf{Z}]$ being produced by this process is:

$$\mathrm{P}\left([\mathbf{Z}]\right) = \exp\left(-\alpha H_{Nm}\right)\left(\frac{\alpha^{K^{+}}}{\prod_{h=0}^{(m+1)^N-1}K_h!}\right)\prod_{k=1}^{K^{+}}\left(\frac{\prod_{n=1}^{N}\binom{m}{z_{nd}}\left(n_k-1\right)!\left(Nm-n_k\right)!}{\left(Nm\right)!}\right),$$

which is the same as the infinite count matrix described in equation (6.2).

Posterior distribution

The posterior distribution for z_{nk} given $\mathbf{Z}_{-(nk)}$ (where $\mathbf{Z}_{-(nk)}$ is the matrix $\mathbf{Z}$ without z_{nk}) is:

$$\begin{aligned}\mathrm{P}\left(z_{nk}|\mathbf{Z}_{-nk}\right) =& \frac{(r!)}{(r-z_{nk})!\,(z_{nk})!}\frac{(n_{-nk}+z_{nk}-1)!\,(Nr-n_{-nk}-z_{nk})!}{(Nr)!}\\ &\times\frac{(Nr-r)!}{(n_{-nk}-1)!\,(Nr-r-m_{-n,d})!},\end{aligned}$$

where $n_{-nk} = \sum_{j \neq n} z_{jk}$ is the number of scoops of dish k that have been taken without the nth customer

$$\begin{aligned}
\mathrm{P}\left(z_{nk}|\mathbf{Z}_{-nk}\right) &= \frac{(m!)}{(m-z_{nk})!\,(z_{nk})!}\frac{(n_{-nk}+z_{nk}-1)!\,(Nr-n_{-nk}-z_{nk})!\,(Nr-m)!}{(Nm)!\,(n_{-nk}-1)!\,(Nr-m-n_{-nk})!} \\
&= \frac{\Gamma\left(m+1\right)}{\Gamma\left(m-z_{nk}+1\right)\Gamma\left(z_{nk}+1\right)}\frac{\Gamma\left(n_{-nk}+z_{nk}\right)\Gamma\left(Nr-n_{-nk}-z_{nk}+1\right)}{\Gamma\left(Nm+1\right)} \\
&\times \frac{\Gamma\left(Nm-m+1\right)}{\Gamma\left(n_{-nk}\right)\Gamma\left(Nm-m-n_{-nk}+1\right)} \\
&= \mathrm{BetaBinomial}\left(n_{-nk},\left(N-1\right)r+1-n_{-nk}\right),
\end{aligned}$$

where $\mathrm{BetaBinomial}\left(\cdot,\cdot\right)$ is the beta-binomial distribution.

6.2 Marked beta-negative-binomial process

6.2.1 Marked beta process

Initially proposed by [7], the *marked beta process* (mBP) is similar to the beta process discussed in section 6.1, however a set-of-three of points $\{(\theta_k, \pi_k, r_k)\}_{k=1}^{\infty}$ are drawn form it; such that $(\theta_k, \pi_k, r_k) \in \mathbf{\Omega} \times [0,1] \times \mathbb{R}^+$ for all k. These points are drawn from Poisson process with the Lévy measure:

$$\nu\left(d\theta, d\pi, dm\right) = \alpha\left(\theta\right)\pi^{-1}\left(1-\pi\right)^{\alpha(\theta)-1} d\pi H_0\left(d\theta\right) R_0\left(dr\right),$$

where $\alpha\left(\cdot\right)$ is a concentration function, and both R_0 & H_0 are continuous measures. Draws from the mBP are:

$$\begin{aligned}
RH|\alpha_0, R_0H_0 \sim& \mathrm{mBP}\left(\alpha_0, R_0H_0\right), \\
RH =& \sum_k \pi_k \delta_{(r_k,\theta_k)},
\end{aligned}$$

where the concentration function $\alpha\left(\cdot\right) = \alpha_0 \in \mathbb{R}^+$ is now a concentration parameter.

6.2.2 Negative-binomial process

Points from the *negative-binomial process* (Neg-BinP) are drawn from Poisson process with the Lévy measure:

$$\lambda\left(d\theta, d\pi, dr\right) = \sum_{i=1}^{\sum_{j=i-1}^{i}\delta_j<r} \delta_i\left(d\pi\right) H\left(d\theta\right) dr,$$

where $\delta_0 = 0$. Note that this is equivalent to continuously sampling from the Bernoulli process (see Section 2.2.2) till r failures have been observed. Draws from the negative-binomial process can be represented as:

$$B_n|RH \sim \text{Neg} - \text{BinP}\left(RH\right),$$

$$B_n = \sum_k b_{nk}\delta_{(r_k,\theta_k)},$$

$$b_{nk}|r_k, \pi_k \sim \text{negative} - \text{binomial}\left(r_k, \pi_k\right),$$

where points $(\theta_k, b_{nk}) \in \mathbf{\Omega} \times \{0, 1, 2, \dots\}$

It is useful to marginalise over the MH and work directly with the posterior marginal process. However, this is impossible as this marginal is not available in closed form. Therefore, for the remainder of the section, we'll focus on a special case of this process, which is the beta-binomial process.

6.2.3 Conjugacy

The marked beta process is conjugate to the negative-binomial process. Let $RH \sim \text{dmBP}\left(\alpha, R_0H_0\right)$ and let $B_{1,\dots,N} \overset{i.i.d.}{\sim} \text{Neg} - \text{BinP}\left(RH\right)$ be N independent samples from the negative-binomial process (with hazard measure RH), then the posterior update on RH is still a marked beta process:

$$RH|\{B_n\}_{n=1}^{N}, \alpha_N, R_NH_N \sim \text{mBP}\left(\alpha_N, R_NH_N\right),$$

where:

$$\begin{aligned}R_N H_N =& \frac{\alpha_0}{\alpha_N} R_0 H_0 + \sum_k \frac{n_k}{\alpha_N} \delta_{(r_k, \theta_k)},\\ \alpha_N(\theta) =& \alpha_0 + n_k + N r_k,\end{aligned}$$

where $n_k = \sum_n b_{nk}$ is the number of times in which the atom at location θ_k appears in $\{B_n\}_{n=1}^N$.

6.2.4 Special case: beta-negative-binomial process

The beta-negative-binomial process is a special case of the marked beta-negative-binomial process from Section 6.2, where the marked beta process is replaced with the beta process.

Beta process A set-of-two of points $\{(\theta_k, \pi_k)\}_{k=1}^{\infty}$ are drawn from beta process; such that $(\theta_k, \pi_k) \in \mathbf{\Omega} \times [0,1]$ for all k. These points are drawn from Poisson process with the Lévy measure:

$$\nu(d\theta, d\pi) = \alpha(\theta) \pi^{-1} (1-\pi)^{\alpha(\theta)-1} d\pi H_0(d\theta),$$

where $\alpha(\cdot)$ is a concentration function, H_0 is a continuous measure. Draws from the BP are:

$$\begin{aligned}H \sim& \text{BP}(\alpha_0, H_0),\\ H =& \sum_k \pi_k \delta_{\theta_k},\end{aligned}$$

where the concentration function $\alpha(\cdot) = \alpha_0 \in \mathbb{R}^+$ is now a concentration parameter.

Negative-binomial process Draws from the negative-binomial process can be represented as:

$$
\begin{aligned}
B_n|mH &\sim \text{Neg} - \text{BinP}\left(mH\right), \\
B_n &= \sum_k b_{nk}\delta_{\theta_k}, \\
b_{nk}|m,\pi_k &\sim \text{negative} - \text{binomial}\left(m,\pi_k\right),
\end{aligned}
$$

where points $(\theta_k, b_{nk}) \in \mathbf{\Omega} \times \{0,1,2,\dots\}$ and $r > 0$.

Conjugacy The beta process is conjugate to the negative-binomial process, let $H|\alpha_0, H_0 \sim \text{BP}\left(\alpha_0, H_0\right)$ and let $\{B_n\}_{n=1}^N|rH \overset{i.i.d.}{\sim} \text{Neg} - \text{BinP}\left(rH\right)$ be N independent samples from the binomial process (with hazard measure rH), then the posterior update on H is still a beta process:

$$
H|\{B_n\}_{n=1}^N, \alpha_N, H_N \sim \text{BP}\left(\alpha_N, H_N\right),
$$

were:

$$
\begin{aligned}
H_N =& \frac{\alpha_0}{\alpha_N} H_0 + \sum_k \frac{n_k}{\alpha_N}\delta_{(\theta_k)}, \\
\alpha_N =& \alpha_0 + n_k + Nr,
\end{aligned}
$$

where $n_k = \sum_n b_{nk}$ is the number of times in which the atom at location θ_k appears in $\{B_n\}_{n=1}^N$.

6.2.5 Distribution on infinite count matrices

In this section, we'll derive the probability distribution over an infinite count matrix.

Finite Let $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]$ be a K-dimensional vector; where $\pi_k \in [0,1]\ \forall k$. The prior distribution over each π_k is the beta distribution:

$$
\pi_k|\alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right),
$$

where $\alpha > 0$ is a hyperparameter. Let **Z** be a $(N \times K)$ where each element $z_{nk} \in \{0, 1, \dots\}$ is independently and identically sampled from a negative-binomial distribution:

$$z_{nk}|m, \pi_k \overset{i.i.d.}{\sim} \text{negative} - \text{binomial}\left(m, \pi_k\right)$$

where $n \in \{1, \dots, N\}$, $k \in \{1, \dots, K\}$, and for simplicity we restrict $r \in \mathbb{N}_0$; although by definition $r > 0$ is also allowed. From a latent feature perspective z_{nk} can be interpreted as the number of times object n possessing feature k. The marginal likelihood of **Z** is:

$$\begin{aligned} \mathrm{P}\left(\mathbf{Z}|\alpha, r\right) &= \prod_{k=1}^{K} \int \left(\prod_{n=1}^{N} \mathrm{P}\left(z_{nk}|\pi_k, r\right) \right) \mathrm{P}\left(\pi_k|\alpha\right) d\pi_k \\ &= \left[\prod_{n=1}^{N} \binom{z_{nk} + r - 1}{z_{nk}} \right] \left(\frac{\alpha}{K} \right) \frac{\Gamma\left(n_k + \frac{\alpha}{K}\right) \Gamma\left(Nr + 1\right)}{\Gamma\left(n_k + \frac{\alpha}{K} + Nr + 1\right)}, \end{aligned} \tag{6.3}$$

where $n_k = \sum_{n=1}^{N} z_{nk}$; see Appendix D.2 for the derivation.

Left order equivalence We use the extended left-ordered form $\mathrm{lof}\left(\cdot\right)$ from Section 6.1.5, however the history for column k is:

$$H_k = \sum_{n=1}^{N} \left(m + 1\right)^{N-n} z_{nk},$$

where m is the maximum value of the matrix Z. The cardinality of $[\mathbf{Z}] = \frac{K!}{\prod_{h=0}^{(m+1)^N - 1} K_h!}$ is the number of matrices that map to the same left-ordered form; where K_h denotes the number of columns with history h.

Infinite count matrix By using the methods from above, the marginal probability of lof-equivalent class of binary matrices [**Z**] is:

$$\begin{aligned} \mathrm{P}\left(\left[\mathbf{Z}\right]|\alpha, m, r\right) &= \sum_{\mathbf{Z} \in [\mathbf{Z}]} \mathrm{P}\left(\mathbf{Z}|\alpha, m\right) \\ &= \frac{K!}{\prod_{h=0}^{(m+1)^N - 1} K_h!} \prod_{k=1}^{K} \left[\prod_{n=1}^{N} \binom{z_{nk} + r - 1}{z_{nk}} \right] \left(\frac{\alpha}{K} \right) \frac{\Gamma\left(n_k + \frac{\alpha}{K}\right) \Gamma\left(Nr + 1\right)}{\Gamma\left(n_k + \frac{\alpha}{K} + Nr + 1\right)}, \end{aligned}$$

and as $K \to \infty$, the above becomes:

$$\begin{aligned} \lim_{K \to \infty} \left(\mathrm{P} \left([\mathbf{Z}] \left| \alpha, m, r \right. \right) \right) = & \frac{\alpha^{K^+}}{\prod_{h=1}^{(m+1)^N - 1} K_h} \\ & \times \exp\left(-\alpha H_{Nr}\right) \prod_{k=1}^{K^+} \frac{\left(\prod_{n=1}^{N} \binom{z_{nk}+r-1}{z_{nk}} \right) (n_k - 1)! \, (Nr)!}{(n_k + Nr)!}, \end{aligned} \tag{6.4}$$

where where $H_{Nr} = \sum_{j=1}^{Nr} \frac{1}{j}$; also known as the (Nr)th harmonic number; see Appendix D.2 for the derivation.

6.2.6 Infinite scoop Indian Buffet Process

In this section, we'll derive the marginal process of the beta-negative-binomial process. We will call this the *Infinite scoop Indian Buffet Process* (isIBP); this is similar to the msIBP (from Section 6.1.6), however unlike the binomial distribution the negative-binomial distribution does not restrict the domain of a random variable, hence we must not restrict the number of scoops in the process. The process will be defined using a cuisine metaphor of an Indian buffet with an infinite number of dishes, where a customer can take any number of scoops of each dish; the dish refers to a feature, and the scoop refers to the number of times a feature is active. This section will be split up into three sub-sections.

New customers

When the first customer enters a buffet, they are given r 'chances' to sample a new dish (which have previously not been sampled); each chance is a sample from a Poisson distribution, the number of new dishes sampled by the first customer on the jth chance is denoted as K_{new}^{nj},which is sampled from:

$$K_{new}^{1j} \sim \text{Poisson}\left(\frac{\alpha}{1r - (r - j)}\right),$$

where $\alpha > 0$ is some constant and $r \in \mathcal{N}_0$. More generally, the r chances for the first customer are:

$$\begin{aligned}
K_{new}^{11} &\sim \text{Poisson}\left(\frac{\alpha}{1r-(r-1)}\right), \\
K_{new}^{12} &\sim \text{Poisson}\left(\frac{\alpha}{1r-(r-2)}\right), \\
&\vdots \\
K_{new}^{1r} &\sim \text{Poisson}\left(\frac{\alpha}{1r}\right).
\end{aligned}$$

The r chances for the nth customer are:

$$\begin{aligned}
K_{new}^{n1} &\sim \text{Poisson}\left(\frac{\alpha}{nr-(r-1)}\right), \\
K_{new}^{n2} &\sim \text{Poisson}\left(\frac{\alpha}{nr-(r-2)}\right), \\
&\vdots \\
K_{new}^{nr} &\sim \text{Poisson}\left(\frac{\alpha}{nr}\right).
\end{aligned}$$

The joint distribution of K_{new}^{ij} for $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, r\}$ is:

$$\begin{aligned}
\mathrm{P}\Big(K_{new}^{11}, ..., K_{new}^{1r}, ..., K_{new}^{N1}, ..., K_{new}^{Nr}\Big) &= \\
&\frac{\exp\left(-\frac{\alpha}{1}\right)\left(\frac{\alpha}{1}\right)^{K_{new}^{11}}}{K_{new}^{11}!} \times ... \times \frac{\exp\left(-\frac{\alpha}{Nr}\right)\left(\frac{\alpha}{Nr}\right)^{K_{new}^{Nr}}}{K_{new}^{Nr}!}.
\end{aligned}$$

The exponent of which is:

$$\begin{aligned}
\exp\left(-\frac{\alpha}{1}\right) \times, .., \times \exp\left(-\frac{\alpha}{Nr}\right) &= \\
&\exp\left(-\alpha\left(1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{Nr}\right)\right) = \exp\left(-\alpha H_{Nr}\right),
\end{aligned}$$

and the total number of dishes sampled are $K^+ = K_{new}^{11} + ... + K_{new}^{1r} + ... + K_{new}^{N1} +$

$... + K_{new}^{Nr}$, which results in:

$$\mathrm{P}\left(K_{new}^{11}, ..., K_{new}^{1r}, ..., K_{new}^{N1}, ..., K_{new}^{Nr}\right) = \exp\left(-\alpha H_{Nm}\right)\left(\frac{\alpha^{K^+}}{\prod_{i=1}^{N}\prod_{j=1}^{r} K_{new}^{ij}!}\right) \times \left(\frac{1}{\prod_{i=1}^{N}\prod_{j=1}^{r}\left(ir-(r-j)\right)^{K_{new}^{ij}}}\right).$$

Existing dishes

Each customer continuously takes scoops from an existing dish (i.e., a dish that has been already sampled) till they fail to take a scoop r number of times. The probability of customer customer n taking a scoop from dish k is $\frac{n_k}{n_k+nr-(r-r_{nk}-1)}$, where n_k is the number of scoops taken from dish k from all customers, and r_{nk} is a count of the number of times customer n has failed to take a scoop from dish k; whereby construction $r_{nk} \in \{0, 1, \ldots, r\}$. This information is stored in a $(N \times K^+)$ count matrix $\mathbf{Z}$; where $z_{nk} \in \{0, 1, \ldots\}$ denotes the number of scoops customer n had of dish k. Then the joint probability of a number of scoops for dish k is:

$$\mathrm{P}\left(z_{1k}, \ldots, z_{Nk}\right) = \left(\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right)\frac{(n_k-1)!\,(Nr)!}{(n_k+Nr)!} \times \left(ir-(r-j)\right),$$

where the left-most component of the product takes the different permutations of customer n into consideration (i.e., for dish k customer n will try z_{nk} scoops out of $z_{nk}+r$ total scoops with the last scoop being failure), and the right-most component is there to offset the fact that customer i first sampled dish k on the jth chance. More generally, for N customers we obtain the following distribution overall K^+ existing dishes:

$$\left(\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right)\frac{(n_k-1)!\,(Nr)!}{(n_k+Nr)!}\right) \times \left(\prod_{i=1}^{N}\prod_{j=1}^{r}\left(ir-(r-j)\right)^{K_{new}^{ij}}\right).$$

Combining existing and new dishes

By combining the results from the two previous sections, the probability of a particular count matrix $\mathbf{Z}$ being produced by this process is:

$$\begin{aligned}\mathrm{P}\left(\mathbf{Z}\right) = \exp\left(-\alpha H_{Nr}\right)&\left(\frac{\alpha^{K^+}}{\prod_{i=1}^{N}\prod_{j=1}^{r}K_{new}^{ij}!}\right)\\ &\times\prod_{k=1}^{K^+}\left(\frac{\prod_{n=1}^{N}\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\left(n_k-1\right)!\left(Nr\right)!}{\left(n_k+Nr\right)!}\right).\end{aligned}$$

As discussed in the section above, the extended $\mathrm{lof}\left(\cdot\right)$ can be used to determine the number of different matrices that encode the same feature assignment; see Figure 6.1. The cardinality of $\left[\mathbf{Z}\right] = \frac{\prod_{i=1}^{N}\prod_{j=1}^{r}K_{new}^{ij}!}{\prod_{h=0}^{(m+1)^N-1}K_h!}$ is the number of matrices that map to the same left-ordered form (i.e., the number of matrices that will encode the same feature assignment); where K_h denotes the number of columns with history h, and m is the maximum value of the matrix $\mathbf{Z}$. Then the probability of a particular feature assignment $\left[\mathbf{Z}\right]$ being produced by this process is:

$$\begin{aligned}\mathrm{P}\left(\left[\mathbf{Z}\right]\right) = \exp\left(-\alpha H_{Nr}\right)&\left(\frac{\alpha^{K^+}}{\prod_{h=0}^{(m+1)^N-1}K_h!}\right)\\ &\times\prod_{k=1}^{K^+}\left(\frac{\prod_{n=1}^{N}\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\left(n_k-1\right)!\left(Nr\right)!}{\left(n_k+Nr\right)!}\right),\end{aligned}$$

which is the same as the infinite count matrix described in equation (6.4).

Posterior distribution

The posterior distribution for z_{nk} given $\mathbf{Z}_{-(nk)}$ (where $\mathbf{Z}_{-(nk)}$ is the matrix $\mathbf{Z}$ without z_{nk}) is:

$$\begin{aligned}\mathrm{P}\left(z_{nk}|\mathbf{Z}_{-nk}\right) &= \left[\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(n_{-nk}+z_{nk}-1\right)!\left(Nr\right)!}{\left(n_{-nk}+z_{nk}+Nr\right)!}\div\frac{\left(n_{-nk}-1\right)!\left(Nr-r\right)!}{\left(n_{-nk}+Nr-r\right)!}\\ &= \left[\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(n_{-nk}+z_{nk}-1\right)!\left(Nr\right)!\left(n_{-nk}+Nr-r\right)!}{\left(n_{-nk}+z_{nk}+Nr\right)!\left(n_{-nk}-1\right)!\left(Nr-r\right)!},\end{aligned}$$

where $n_{-nk} = \sum_{j \neq n} z_{jk}$ is the number of scoops of dish k that have been taken without the nth customer

$$\begin{aligned}
\mathrm{P}\left(z_{nd} | \mathbf{Z}_{-nk}\right) &= \left[\binom{z_{nk}+r-1}{z_{nk}}\right] \frac{\left(n_{-nk}+z_{nk}-1\right)!\,(Nr)!\,\left(n_{-nk}+Nr-r\right)!}{\left(n_{-nk}+z_{nk}+Nr\right)!\,\left(n_{-nk}-1\right)!\,(Nr-r)!} \\
&= \left[\binom{z_{nk}+r-1}{z_{nk}}\right] \frac{\Gamma\left(n_{-nk}+z_{nk}\right)\Gamma\left(Nr+1\right)\Gamma\left(m_{-n,d}+Nr-r+1\right)}{\Gamma\left(n_{-nk}+z_{nk}+Nr+1\right)\Gamma\left(n_{-nk}\right)\Gamma\left(Nr-r+1\right)} \\
&= \text{BetaNegative} - \text{binomial}\left(Nr-r+1, n_{-nk}\right),
\end{aligned}$$

where BetaNegative − binomial $(\cdot, \cdot)$ is the beta-negative-binomial distribution.

6.3 Discussion

In this chapter, we proposed two novel processes The first process is the discrete marked beta-binomial process, its special case the beta-binomial process, and its marginal multi-scoop IBP (msIBP). This process generalises the work done by [8, 19]. The msIBP can be used as a prior on the latent space if it is assumed that the latent space consists of counts which are restricted to some value, i.e., the count of a feature cannot be more than $m \in \mathbb{N}$.

The marked beta-negative-binomial process builds upon the work proposed by [7, 19], where its special case the beta-negative-binomial process, and its infinite scoop IBP (isIBP) are novel. The isIBP can be used as a prior on the latent space if it is assumed that the latent space consists of counts which are not restricted to some value, i.e., the count of a feature has no upper bound.

One extension of this work is to discover the marginal processes of the actual processes, i.e., the marginal of the discrete marked beta-binomial and marked beta-negative-binomial process.

Chapter 7

Summary and future work

In this thesis, we presented novel latent variable models to solve unsupervised problems like clustering and dimensionality reduction.

In Chapter 3 we proposed a novel mixture model for modelling (in terms of density estimation or clustering) count data; we call it the Panjer mixture. Unlike other mixture models, the Panjer mixture model makes no a-priori assumption about the dispersion of the data, which results in better clustering and density estimation. One promising extension of this work is to propose a multivariate extension of the Panjer mixture model.

In Chapter 4 we propose a novel approach to robustify models with respect to any potential likelihood misspesification by using pseudo-points to represent it; this is done by using the MMD. The proposed method is applied to mixture models, we call this the mixtures of maximum mean discrepancy pseudo-point marginal, a simple EM scheme is also presented to infer the parameters. One promising future direction of the proposed approach is to apply this in other domains such as dimensionality reduction or training variational autoencoders.

In Chapter 5 we proposed two novel dimensionality reduction techniques which utilise constrained feature allocation induced by the multivariate hypergeometric distribution; this overcomes the over-partitioning obtained from models which utilise the beta processes [85]. The use of constrained feature allocation can be applied to models like the Gaussian process latent variable model (GPLVMs), this will allow for a richer and more compact model of the manifold learning using a smaller number of underlying, feature-specific Gaussian processes.

In Chapter 6 we proposed two novel discrete Bayesian nonparametric priors which generalise existing Bayesian nonparametric priors. Furthermore, the marginal process of the special cases was also derived to assist in sampling from these priors. Chapter 6 only derives these processes, and therefore a potential extension of this work is to apply these priors in applications where the latent space consists of counts; for example non-negative matrix factorisation of count data, or clustering where the count of the latent variable controls for the level of 'intensity'.

Bibliography

[1] Jing Wu and Zheng Lin. "Research on customer segmentation model by clustering." In: *Proceedings of the 7th international conference on Electronic commerce.* 2005, pp. 316–318.

[2] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[3] James MacQueen et al. "Some methods for classification and analysis of multivariate observations." In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.* Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[4] Adam Farooq et al. "Controlling for sparsity in sparse factor analysis models: adaptive latent feature sharing for piecewise linear dimensionality reduction." In: *arXiv preprint arXiv:2006.12369* (2020).

[5] Yee W Teh and Dilan Gorur. "Indian buffet processes with power-law behavior." In: *Advances in neural information processing systems.* 2009, pp. 1838–1846.

[6] Romain Thibaux and Michael I Jordan. "Hierarchical beta processes and the Indian buffet process." In: *Artificial Intelligence and Statistics.* 2007, pp. 564–571.

[7] Mingyuan Zhou et al. "Beta-negative binomial process and Poisson factor analysis." In: *Artificial Intelligence and Statistics.* PMLR. 2012, pp. 1462–1471.

[8] Zoubin Ghahramani and Thomas L Griffiths. "Infinite latent feature models and the Indian buffet process." In: *Advances in neural information processing systems.* 2006, pp. 475–482.

[9] PS Laplace. “Essai Philosophique sur les Probabilities, Courcier Imprimeur, Paris; reprints of this work and of Laplace’s much larger Theorie Analytic des Probabilities are available from Editions Culture et Civilisation, 115 Ave.” In: *Cabriel Lebron* 1160 (1814).

[10] Edwin James George Pitman. “Sufficient statistics and intrinsic accuracy.” In: *Mathematical Proceedings of the cambridge Philosophical society*. Vol. 32. 4. Cambridge University Press. 1936, pp. 567–579.

[11] Morris DeGroot. *H., Optimal Statistical Decisions*. 1970.

[12] Kevin P Murphy. “Conjugate Bayesian analysis of the Gaussian distribution.” In: *def* $1.2\sigma 2$ (2007), p. 16.

[13] Chris Fraley and Adrian E Raftery. “Model-based clustering, discriminant analysis, and density estimation.” In: *Journal of the American statistical Association* 97.458 (2002), pp. 611–631.

[14] Gideon Schwarz. “Estimating the dimension of a model.” In: *The annals of statistics* (1978), pp. 461–464.

[15] Thomas S Ferguson. “A Bayesian analysis of some nonparametric problems.” In: *The annals of statistics* (1973), pp. 209–230.

[16] Nils Lid Hjort et al. “Nonparametric Bayes estimators based on beta processes in models for life history data.” In: *The Annals of Statistics* 18.3 (1990), pp. 1259–1294.

[17] JM Bernardo and AFM Smith. “Bayesian theory wiley.” In: *New York* 49 (1994).

[18] David J Aldous. “Exchangeability and related topics.” In: *École d’Été de Probabilités de Saint-Flour XIII—1983*. Springer, 1985, pp. 1–198.

[19] Romain Jean Thibaux. *Nonparametric Bayesian models for machine learning*. University of California, Berkeley, 2008.

[20] Michael I Jordan. “Hierarchical models, nested models and completely random measures.” In: *Frontiers of statistical decision making and Bayesian analysis: In honor of James O. Berger. New York: Springer* (2010), pp. 207–218.

[21] Charmaine Dean, JF Lawless, and GE Willmot. “A mixed poisson–inverse-gaussian regression model.” In: *Canadian Journal of Statistics* 17.2 (1989), pp. 171–181.

[22] Dimitris Karlis. “An EM algorithm for multivariate Poisson distribution and related models.” In: *Journal of Applied Statistics* 30.1 (2003), pp. 63–77.

[23] Geng Chen et al. “Single-cell analyses of X Chromosome inactivation dynamics and pluripotency during differentiation.” In: *Genome research* 26.10 (2016), pp. 1342–1354.

[24] Dominic Grün et al. “Single-cell messenger RNA sequencing reveals rare intestinal cell types.” In: *Nature* 525.7568 (2015), pp. 251–255.

[25] Malte D Luecken and Fabian J Theis. “Current best practices in single-cell RNA-seq analysis: a tutorial.” In: *Molecular systems biology* 15.6 (2019), e8746.

[26] JP Meador et al. “Massively parallel single-cell.” In: *Science* 343 (2014), pp. 776–779.

[27] Junyue Cao et al. “Comprehensive single-cell transcriptional profiling of a multicellular organism.” In: *Science* 357.6352 (2017), pp. 661–667.

[28] Thomas Hofmann. “Probabilistic latent semantic analysis.” In: *arXiv preprint arXiv:1301.6705* (2013).

[29] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation.” In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[30] Richard W Conway and William L Maxwell. “A queuing model with state dependent service rates.” In: *Journal of Industrial Engineering* 12.2 (1962), pp. 132–136.

[31] Carl M Harris. “On finite mixtures of geometric and negative binomial distributions.” In: *Communications in Statistics-Theory and Methods* 12.9 (1983), pp. 987–1007.

[32] Dimitris Karlis and Loukia Meligkotsidou. “Finite mixtures of multivariate Poisson distributions with application.” In: *Journal of statistical Planning and Inference* 137.6 (2007), pp. 1942–1960.

[33] Kenneth W Church and William A Gale. “Poisson mixtures.” In: *Natural Language Engineering* 1.2 (1995), pp. 163–190.

[34] Geurt Jongbloed and Ger Koole. “Managing uncertainty in call centres using Poisson mixtures.” In: *Applied Stochastic Models in Business and Industry* 17.4 (2001), pp. 307–318.

[35] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. “The concrete distribution: A continuous relaxation of discrete random variables.” In: *arXiv preprint arXiv:1611.00712* (2016).

[36] Bent Jorgensen. *Statistical properties of the generalized inverse Gaussian distribution.* Vol. 9. Springer Science & Business Media, 2012.

[37] Charmaine Dean, JF Lawless, and GE Willmot. “A mixed poisson–inverse-gaussian regression model.” In: *Canadian Journal of Statistics* 17.2 (1989), pp. 171–181.

[38] Rolf Luders. “Die statistik der seltenen ereignisse.” In: *Biometrika* 26.1/2 (1934), pp. 108–128.

[39] Robert A Rigby, Dimitrios M Stasinopoulos, and Calliope Akantziliotou. “A framework for modelling overdispersed count data, including the Poisson-shifted generalized inverse Gaussian distribution.” In: *Computational Statistics & Data Analysis* 53.2 (2008), pp. 381–393.

[40] Thomas P Minka et al. “Computing with the COM-Poisson distribution.” In: *PA: Department of* 776 (2003).

[41] Charalampos Chanialidis. "Bayesian mixture models for count data." PhD thesis. University of Glasgow, 2015.

[42] Bjørn Sundt and William S Jewell. "Further results on recursive evaluation of compound distributions." In: *ASTIN Bulletin: The Journal of the IAA* 12.1 (1981), pp. 27–39.

[43] Harry H Panjer. "Recursive evaluation of a family of compound distributions." In: *ASTIN Bulletin: The Journal of the IAA* 12.1 (1981), pp. 22–26.

[44] Michael Fackler. "Panjer class revisited: one formula for the distributions of the Panjer (a, b, n) class." In: *Available at SSRN 3813246* (2021).

[45] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[46] Marina Meilă. "Comparing clusterings—an information based distance." In: *Journal of multivariate analysis* 98.5 (2007), pp. 873–895.

[47] Sara Wade, Zoubin Ghahramani, et al. "Bayesian cluster analysis: Point estimation and credible balls (with discussion)." In: *Bayesian Analysis* 13.2 (2018), pp. 559–626.

[48] Marina Meilă. "Comparing clusterings by the variation of information." In: *Learning theory and kernel machines*. Springer, 2003, pp. 173–187.

[49] Sarah Caul. *Office for National Statistics: Deaths registered weekly in England and Wales, provisional.* Link here. [Online; accessed 17-August-2021]. 2021.

[50] LHC Tippett. "Technological Applications of." In: *Statistics. John Wiley & Sons, Inc., c* (1950).

[51] Jean-Patrick Baudry et al. "Combining mixture components for clustering." In: *Journal of computational and graphical statistics* 19.2 (2010), pp. 332–353.

[52] Robert A Jacobs et al. "Adaptive mixtures of local experts." In: *Neural computation* 3.1 (1991), pp. 79–87.

[53] Matthew J Johnson et al. "Composing graphical models with neural networks for structured representations and fast inference." In: *Advances in neural information processing systems* 29 (2016), pp. 2946–2954.

[54] Antonio Lijoi, Igor Prünster, and Tommaso Rigon. "Finite-dimensional discrete random structures and Bayesian clustering." In: *Preprint* (2020).

[55] Katherine A Heller and Zoubin Ghahramani. "A nonparametric bayesian approach to modeling overlapping clusters." In: *Artificial Intelligence and Statistics*. PMLR. 2007, pp. 187–194.

[56] Sally Paganin et al. "Centered Partition Processes: Informative Priors for Clustering (with Discussion)." In: *Bayesian Analysis* 16.1 (2021), pp. 301–370.

[57] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[58] Max Welling and Kenichi Kurihara. "Bayesian K-means as a "maximization-expectation" algorithm." In: *Proceedings of the 2006 SIAM international conference on data mining*. SIAM. 2006, pp. 474–478.

[59] Michael C Hughes and Erik B Sudderth. *Memoized online variational inference for Dirichlet process mixture models*. Tech. rep. BROWN UNIV PROVIDENCE RI DEPT OF COMPUTER SCIENCE, 2014.

[60] Alex Tank, Nicholas Foti, and Emily Fox. "Streaming variational inference for Bayesian nonparametric mixture models." In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 968–976.

[61] Lianming Wang and David B Dunson. "Fast Bayesian inference in Dirichlet process mixture models." In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 196–216.

[62] Yordan P Raykov, Alexis Boukouvalas, Max A Little, et al. "Simple approximate MAP inference for Dirichlet processes mixtures." In: *Electronic Journal of Statistics* 10.2 (2016), pp. 3548–3578.

[63] Sylvia Frühwirth-Schnatter. “Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models.” In: *Journal of the American Statistical Association* 96.453 (2001), pp. 194–209.

[64] Radford M Neal. “Markov chain sampling methods for Dirichlet process mixture models.” In: *Journal of computational and graphical statistics* 9.2 (2000), pp. 249–265.

[65] Sinead Williamson, Avinava Dubey, and Eric Xing. “Parallel Markov chain Monte Carlo for nonparametric mixture models.” In: *International Conference on Machine Learning*. PMLR. 2013, pp. 98–106.

[66] Syed Mumtaz Ali and Samuel D Silvey. “A general class of coefficients of divergence of one distribution from another.” In: *Journal of the Royal Statistical Society: Series B (Methodological)* 28.1 (1966), pp. 131–142.

[67] Bharath K Sriperumbudur et al. “On integral probability metrics,\phi-divergences and binary classification.” In: *arXiv preprint arXiv:0901.2698* (2009).

[68] Alfred Müller. “Integral probability metrics and their generating classes of functions.” In: *Advances in Applied Probability* 29.2 (1997), pp. 429–443.

[69] Badr-Eddine Chérief-Abdellatif and Pierre Alquier. “MMD-Bayes: Robust Bayesian estimation via maximum mean discrepancy.” In: *Symposium on Advances in Approximate Bayesian Inference*. PMLR. 2020, pp. 1–21.

[70] Francois-Xavier Briol et al. “Statistical inference for generative models with maximum mean discrepancy.” In: *arXiv preprint arXiv:1906.05944* (2019).

[71] Arthur Gretton et al. “A kernel two-sample test.” In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[72] John Shawe-Taylor and Robert C Williamson. “A PAC analysis of a Bayesian estimator.” In: *Proceedings of the tenth annual conference on Computational learning theory*. 1997, pp. 2–9.

[73] David A McAllester. “Some pac-bayesian theorems.” In: *Machine Learning* 37.3 (1999), pp. 355–363.

[74] Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. “A general framework for updating belief distributions.” In: *Journal of the Royal Statistical Society. Series B, Statistical methodology* 78.5 (2016), p. 1103.

[75] Joseph G Ibrahim and Ming-Hui Chen. “Power prior distributions for regression models.” In: *Statistical Science* (2000), pp. 46–60.

[76] Jeffrey W Miller and David B Dunson. “Robust Bayesian inference via coarsening.” In: *Journal of the American Statistical Association* (2018).

[77] Jack Jewson, Jim Q Smith, and Chris Holmes. “Principled Bayesian minimum divergence inference.” In: *Preprint* (2018).

[78] SP Lyddon, SG Walker, and Chris C Holmes. “Nonparametric learning from Bayesian models with randomized objective functions.” In: *arXiv preprint arXiv:1806.11544* (2018).

[79] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python.” In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[80] 10x Genomics. *500 Peripheral blood mononuclear cells (PBMCs) from a healthy donor (Next GEM v1.1).* [Online; accessed 17-Sep-2021].

[81] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction.” In: *arXiv preprint arXiv:1802.03426* (2018).

[82] Harry H Harman. *Modern factor analysis.* Univ. of Chicago Press, 1960.

[83] Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space.” In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[84] Barbara E Engelhardt and Matthew Stephens. “Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis.” In: *PLoS Genet* 6.9 (2010), e1001117.

[85] David Knowles and Zoubin Ghahramani. “Infinite sparse factor analysis and infinite independent components analysis.” In: *International Conference on Independent Component Analysis and Signal Separation*. Springer. 2007, pp. 381–388.

[86] John Paisley and Lawrence Carin. “Nonparametric factor analysis with beta process priors.” In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 777–784.

[87] Anirban Bhattacharya and David B Dunson. “Sparse Bayesian infinite factor models.” In: *Biometrika* (2011), pp. 291–306.

[88] Chuan Gao, Christopher D Brown, and Barbara E Engelhardt. “A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects.” In: *arXiv preprint arXiv:1310.4792* (2013).

[89] David A Van Dyk and Taeyoung Park. “Partially collapsed Gibbs samplers: Theory and methods.” In: *Journal of the American Statistical Association* 103.482 (2008), pp. 790–796.

[90] George Kingsley Zipf. “Selected studies of the principle of relative frequency in language.” In: (1932).

[91] Lianming Wang and David B Dunson. “Fast Bayesian inference in Dirichlet process mixture models.” In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 196–216.

[92] Tamara Broderick, Brian Kulis, and Michael Jordan. “MAD-Bayes: MAP-based asymptotic derivations from Bayes.” In: *International Conference on Machine Learning*. 2013, pp. 226–234.

[93] Yordan Raykov. “A deterministic inference framework for discrete nonparametric latent variable models: learning complex probabilistic models with simple algorithms.” PhD thesis. Aston University, 2017.

[94] Jeffrey W Miller and Matthew T Harrison. “A simple example of Dirichlet process mixture inconsistency for the number of components.” In: *Advances in neural information processing systems*. 2013, pp. 199–206.

[95] Li Deng. “The mnist database of handwritten digit images for machine learning research [best of the web].” In: *IEEE signal processing magazine* 29.6 (2012), pp. 141–142.

[96] John Paisley and Lawrence Carin. “Nonparametric factor analysis with beta process priors.” In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 777–784.

[97] Ian T Jolliffe and Jorge Cadima. “Principal component analysis: a review and recent developments.” In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[98] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[99] Hui Zou, Trevor Hastie, and Robert Tibshirani. “Sparse principal component analysis.” In: *Journal of computational and graphical statistics* 15.2 (2006), pp. 265–286.

[100] Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. “A modified principal component technique based on the LASSO.” In: *Journal of computational and Graphical Statistics* 12.3 (2003), pp. 531–547.

[101] Andrew Gelman et al. *Bayesian data analysis*. CRC press, 2013.

[102] Bob Carpenter et al. “Stan: A probabilistic programming language.” In: *Journal of statistical software* 76.1 (2017).

[103] Akihiko Nishimura, David Dunson, and Jianfeng Lu. “Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods.” In: *arXiv preprint arXiv:1705.08510* (2017).

[104] Adrian E Raftery and Steven M Lewis. “[Practical Markov Chain Monte Carlo]: comment: one long run with diagnostics: implementation strategies for Markov Chain Monte Carlo.” In: *Statistical science* 7.4 (1992), pp. 493–497.

[105] Hemant D Tagare. “Notes on optimization on stiefel manifolds.” In: *Technical report, Technical report.* Yale University, 2011.

[106] Clément Elvira, Pierre Chainais, and Nicolas Dobigeon. “Bayesian nonparametric Principal Component Analysis.” In: *arXiv preprint arXiv:1709.05667* (2017).

[107] Christopher Bingham. “An antipodally symmetric distribution on the sphere.” In: *The Annals of Statistics* (1974), pp. 1201–1225.

[108] CG Khatri and Kanti V Mardia. “The von Mises–Fisher matrix distribution in orientation statistics.” In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 95–106.

[109] Carl S Herz. “Bessel functions of matrix argument.” In: *Annals of Mathematics* (1955), pp. 474–523.

[110] Christopher J Fallaize and Theodore Kypraios. “Exact bayesian inference for the bingham distribution.” In: *Statistics and Computing* 26.1-2 (2016), pp. 349–360.

[111] James Townsend, Niklas Koep, and Sebastian Weichwald. “Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation.” In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 4755–4759.

[112] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes.” In: *arXiv preprint arXiv:1312.6114* (2013).

[113] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. “Columbia object image library (coil-100).” In: (1996).

[114] Eric Zarahn, Geoffrey K Aguirre, and Mark D’Esposito. “Empirical analyses of BOLD fMRI statistics.” In: *NeuroImage* 5.3 (1997), pp. 179–197.

[115] Raimon HR Pruim et al. “ICA-AROMA: A robust ICA-based strategy for removing motion artifacts from fMRI data.” In: *Neuroimage* 112 (2015), pp. 267–277.

[116] Vince D Calhoun et al. “ICA of functional MRI data: an overview.” In: *in Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation.* Citeseer. 2003.

[117] Jalil Taghia et al. “Bayesian switching factor analysis for estimating time-varying functional connectivity in fMRI.” In: *Neuroimage* 155 (2017), pp. 271–290.

[118] Christian F Beckmann and Stephen M Smith. “Tensorial extensions of independent component analysis for multisubject FMRI analysis.” In: *Neuroimage* 25.1 (2005), pp. 294–311.

[119] Pedro AdFR Højen-Sørensen, Ole Winther, and Lars Kai Hansen. “Analysis of functional neuroimages using ICA with adaptive binary sources.” In: *Neurocomputing* 49.1-4 (2002), pp. 213–225.

[120] Martin J McKeown, Lars Kai Hansen, and Terrence J Sejnowsk. “Independent component analysis of functional MRI: what is signal and what is noise?” In: *Current opinion in neurobiology* 13.5 (2003), pp. 620–629.

[121] Vince D Calhoun, Jingyu Liu, and Tülay Adalı. “A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data.” In: *Neuroimage* 45.1 (2009), S163–S172.

[122] Zhenwen Dai, James Hensman, and Neil Lawrence. “Spike and slab gaussian process latent variable models.” In: *arXiv preprint arXiv:1505.02434* (2015).

[123] Yordan P Raykov et al. “What to do when k-means clustering fails: A simple yet principled alternative algorithm.” In: *PloS one* 11.9 (2016), e0162259.

[124] Adam Farooq et al. *Adaptive Probabilistic Principal Components Analysis. In NIPS 2018 Workshop: All of Bayesian Nonparametrics.* 2018.

[125] Nilesh Tripuraneni et al. “Particle Gibbs for Infinite Hidden Markov Models.” In: *Advances in Neural Information Processing Systems.* Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: `https://proceedings.neurips.cc/paper/2015/file/4edaa105d5f53590338791951e38c3ad-Paper.pdf`.

[126] Rens van de Schoot et al. “Bayesian statistics and modelling.” In: *Nature Reviews Methods Primers* 1.1 (2021), pp. 1–26.

[127] Bridget E Weller, Natasha K Bowen, and Sarah J Faubert. “Latent class analysis: a guide to best practice.” In: *Journal of Black Psychology* 46.4 (2020), pp. 287–311.

[128] Edward Meeds et al. “Modeling dyadic data with binary latent factors.” In: *Advances in neural information processing systems* 19 (2006).

[129] Daniele Durante. “A note on the multiplicative gamma process.” In: *Statistics & Probability Letters* 122 (2017), pp. 198–204.

[130] Sirio Legramanti, Daniele Durante, and David B Dunson. “Bayesian cumulative shrinkage for infinite factorizations.” In: *Biometrika* 107.3 (2020), pp. 745–752.

[131] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[132] Yann LeCun, Corinna Cortes, and Christopher JC Burges. “The MNIST database of handwritten digits, 1998.” In: *URL http://yann. lecun. com/exdb/mnist* 10 (1998), p. 34.

[133] Zoubin Ghahramani, Geoffrey E Hinton, et al. *The EM algorithm for mixtures of factor analyzers*. Tech. rep. Technical Report CRG-TR-96-1, University of Toronto, 1996.

[134] Walid M Abdelmoula et al. “Interactive visual exploration of 3D mass spectrometry imaging data using hierarchical stochastic neighbor embedding reveals spatiomolecular structures at full data resolution.” In: *Journal of proteome research* 17.3 (2018), pp. 1054–1064.

[135] Nicole V Acuff and Joel Linden. “Using visualization of t-distributed stochastic neighbor embedding to identify immune cell subsets in mouse tumors.” In: *The Journal of Immunology* 198.11 (2017), pp. 4539–4546.

[136] Julian Besag. “On the statistical analysis of dirty pictures.” In: *Journal of the Royal Statistical Society: Series B (Methodological)* 48.3 (1986), pp. 259–279.

[137] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. “GTM: The generative topographic mapping.” In: *Neural computation* 10.1 (1998), pp. 215–234.

[138] Tamara Broderick et al. “Combinatorial clustering and the beta negative binomial process.” In: *IEEE transactions on pattern analysis and machine intelligence* 37.2 (2014), pp. 290–306.

[139] Tamara Broderick, Ashia C Wilson, Michael I Jordan, et al. “Posteriors, conjugacy, and exponential families for completely random measures.” In: *Bernoulli* 24.4B (2018), pp. 3181–3221.

[140] Minhua Chen et al. “Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds.” In: *IEEE Transactions on Signal Processing* 58.12 (2010), pp. 6140–6155.

[141] Minhua Chen et al. “Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds.” In: *IEEE Transactions on Signal Processing* 58.12 (2010), pp. 6140–6155.

[142] Pierre Comon. “Independent component analysis, a new concept?” In: *Signal processing* 36.3 (1994), pp. 287–314.

[143] P Damlen, John Wakefield, and Stephen Walker. “Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.2 (1999), pp. 331–344.

[144] Giuseppe Di Benedetto, François Caron, and Yee Whye Teh. “Non-exchangeable feature allocation models with sublinear growth of the feature sizes.” In: *arXiv preprint arXiv:2003.13491* (2020).

[145] David L Donoho and Carrie Grimes. “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data.” In: *Proceedings of the National Academy of Sciences* 100.10 (2003), pp. 5591–5596.

[146] Charles W Fox and Stephen J Roberts. “A tutorial on variational Bayesian inference.” In: *Artificial intelligence review* 38.2 (2012), pp. 85–95.

[147] Karl Friston. “Causal modelling and brain connectivity in functional magnetic resonance imaging.” In: *PLoS biology* 7.2 (2009), e1000033.

[148] Dilan Görür, Frank Jäkel, and Carl Edward Rasmussen. "A choice model with infinitely many latent features." In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 361–368.

[149] Zoubin Ghahramani. "Factorial learning and the EM algorithm." In: *Advances in neural information processing systems*. 1995, pp. 617–624.

[150] Zoubin Ghahramani, Thomas L Griffiths, and Peter Sollich. "Bayesian nonparametric latent feature models." In: (2007).

[151] Michael Hanke et al. "A high-resolution 7-Tesla fMRI dataset from complex natural stimulation with an audio movie." In: *Scientific data* 1 (2014), p. 140003.

[152] Katherine A Heller and Zoubin Ghahramani. "A nonparametric bayesian approach to modeling overlapping clusters." In: *Artificial Intelligence and Statistics*. 2007, pp. 187–194.

[153] Geoffrey E Hinton and Sam T Roweis. "Stochastic neighbor embedding." In: *Advances in neural information processing systems*. 2003, pp. 857–864.

[154] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks." In: *science* 313.5786 (2006), pp. 504–507.

[155] Mingwei Huang, Zhen Wang, and Zilu Ying. "Facial expression recognition using stochastic neighbor embedding and SVMs." In: *Proceedings 2011 International Conference on System Science and Engineering*. IEEE. 2011, pp. 671–674.

[156] Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. "Mixtures of robust probabilistic principal component analyzers." In: *Neurocomputing* 71.7-9 (2008), pp. 1274–1282.

[157] Michael C Hughes and Erik Sudderth. "Memoized online variational inference for Dirichlet process mixture models." In: *Advances in Neural Information Processing Systems*. 2013, pp. 1133–1141.

[158] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications." In: *Neural networks* 13.4-5 (2000), pp. 411–430.

[159] Hemant Ishwaran and Lancelot F James. "Gibbs sampling methods for stick-breaking priors." In: *Journal of the American Statistical Association* 96.453 (2001), pp. 161–173.

[160] Ke Jiang, Brian Kulis, and Michael I Jordan. "Small-variance asymptotics for exponential family Dirichlet process mixture models." In: *Advances in Neural Information Processing Systems*. 2012, pp. 3158–3166.

[161] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. "Density-connected subspace clustering for high-dimensional data." In: *Proceedings of the 2004 SIAM international conference on data mining*. SIAM. 2004, pp. 246–256.

[162] Eric Schmitt and Kaveh Vakili. "The FastHCS algorithm for robust PCA." In: *Statistics and Computing* 26 (Sept. 2015), pp. 1229–1242. DOI: 10.1007/s11222-015-9602-5.

[163] Yongdai Kim. "Nonparametric Bayesian estimators for counting processes." In: *Annals of Statistics* (1999), pp. 562–588.

[164] Diederik P Kingma and Max Welling. "Auto-encoding variational Bayes." In: *arXiv preprint arXiv:1312.6114* (2013).

[165] JFC Kingman. "Poisson Processes Oxford University Press." In: *O xford* (1993).

[166] John Kingman. "Completely random measures." In: *Pacific Journal of Mathematics* 21.1 (1967), pp. 59–78.

[167] Plamen Koev and Alan Edelman. "The efficient evaluation of the hypergeometric function of a matrix argument." In: *Mathematics of Computation* 75.254 (2006), pp. 833–846.

[168] Tamio Koyama et al. "Holonomic graient escent for the Fisher-Bingham Distribution on the d-dimensional sphere." In: *Computational Statistics* 29.3-4 (2014), pp. 661–683.

[169] Zoubin Ghahramani, Thomas L Griffiths, and Peter Sollich. "Bayesian nonparametric latent feature models." In: ().

[170] S Mohan Kumar and G Balakrishnan. “Classification of Microcalcification in Digital Mammogram using Stochastic Neighbor Embedding and KNN Classifier.” In: *International Conference on Emerging Technology Trends on Advanced Engineering Research (ICETT’12)*. 2013.

[171] Alfred Kume and Andrew TA Wood. “Saddlepoint approximations for the Bingham and Fisher–Bingham normalising constants.” In: *Biometrika* 92.2 (2005), pp. 465–476.

[172] Gerhard Kurz et al. “Recursive Bingham filter for directional estimation involving 180 degree symmetry.” In: *Journal of Advances in Information Fusion* 9.2 (2014), pp. 90–105.

[173] Neil D Lawrence. “Gaussian process latent variable models for visualisation of high dimensional data.” In: *Advances in neural information processing systems*. 2004, pp. 329–336.

[174] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[175] Radford M Neal et al. “Slice sampling.” In: *The annals of statistics* 31.3 (2003), pp. 705–767.

[176] Jean Chesson. “A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation.” In: *Journal of Applied Probability* 13.4 (1976), pp. 795–797.

[177] Vinayak Rao and Yee W Teh. “Spatial normalized gamma processes.” In: *Advances in neural information processing systems*. 2009, pp. 1554–1562.

[178] Carl Edward Rasmussen. “Gaussian processes in machine learning.” In: *Summer School on Machine Learning*. Springer. 2003, pp. 63–71.

[179] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Kernel principal component analysis.” In: *International Conference on Artificial Neural Networks*. Springer. 1997, pp. 583–588.

[180] Jayaram Sethuraman. “A constructive definition of Dirichlet priors.” In: *Statistica sinica* (1994), pp. 639–650.

[181] Yee Whye Teh. “Dirichlet process.” In: *Encyclopedia of machine learning* (2010), pp. 280–287.

[182] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. “Stick-breaking construction for the Indian buffet process.” In: *Artificial Intelligence and Statistics*. 2007, pp. 556–563.

[183] Joshua B Tenenbaum, Vin De Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction.” In: *science* 290.5500 (2000), pp. 2319–2323.

[184] Michael E Tipping and Christopher M Bishop. “Mixtures of probabilistic principal component analyzers.” In: *Neural computation* 11.2 (1999), pp. 443–482.

[185] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[186] Michalis K Titsias. “The infinite gamma-Poisson feature model.” In: *Advances in Neural Information Processing Systems*. 2008, pp. 1513–1520.

[187] Vincent van Unen et al. “Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types.” In: *Nature communications* 8.1 (2017), p. 1740.

[188] Vincent G van de Ven et al. “Functional connectivity as revealed by spatial independent component analysis of fMRI measurements during rest.” In: *Human brain mapping* 22.3 (2004), pp. 165–178.

[189] Yining Wang and Jun Zhu. “DP-space: Bayesian nonparametric subspace clustering with small-variance asymptotics.” In: *International Conference on Machine Learning*. 2015, pp. 862–870.

[190] Sinead Williamson. *Completely random measures and related models.* http://cbl.eng.cam.ac.uk/pub/Intranet/MLG/ReadingGroup/crm.pdf. Jan. 2011.

[191] Frank Wood and Thomas L Griffiths. “Particle filtering for nonparametric Bayesian matrix factorization.” In: *Advances in neural information processing systems.* 2007, pp. 1513–1520.

[192] Mingyuan Zhou et al. “Beta-negative binomial process and Poisson factor analysis.” In: *Artificial Intelligence and Statistics.* 2012, pp. 1462–1471.

[193] *Spiegel Online. 2008. The World from Berlin: ‘Choosing Beijing Was a Drastic Mistake’.* http://www.spiegel.de/international/world/the-world-from-berlin-choosing-beijing-was-a-drastic-mistake-a-569903.html. Aug. 2008.

[194] Giuseppe Di Benedetto, François Caron, and Yee Whye Teh. “Non-exchangeable feature allocation models with sublinear growth of the feature sizes.” In: *arXiv preprint arXiv:2003.13491* (2020).

[195] Dougal Maclaurin, David Duvenaud, and Ryan P Adams. “Autograd: Effortless gradients in numpy.” In: *ICML 2015 AutoML Workshop.* Vol. 238. 2015.

[196] Arthur Asuncion and David Newman. *UCI machine learning repository.* 2007.

[197] Hélène Jacqmin-Gadda et al. “Robustness of the linear mixed model to misspecified error distribution.” In: *Computational Statistics & Data Analysis* 51.10 (2007), pp. 5142–5154.

[198] Piyush Rai and Hal Daumé. “The infinite hierarchical factor regression model.” In: *Advances in Neural Information Processing Systems.* 2009, pp. 1321–1328.

[199] Sam Roweis and Zoubin Ghahramani. “A unifying review of linear Gaussian models.” In: *Neural computation* 11.2 (1999), pp. 305–345.

[200] JM Bernardo et al. “Bayesian factor regression models in the “large p, small n” paradigm.” In: *Bayesian statistics* 7 (2003), pp. 733–742.

[201] Kieran R Campbell and Christopher Yau. “Probabilistic modeling of bifurcations in single-cell gene expression data using a Bayesian mixture of factor analyzers.” In: *Wellcome open research* 2 (2017).

[202] René Vidal. “Subspace clustering.” In: *IEEE Signal Processing Magazine* 28.2 (2011), pp. 52–68.

[203] Lance Parsons, Ehtesham Haque, and Huan Liu. “Subspace clustering for high dimensional data: a review.” In: *Acm Sigkdd Explorations Newsletter* 6.1 (2004), pp. 90–105.

[204] Harry H Harman. *Modern factor analysis.* University of Chicago press, 1976.

[205] Zoubin Ghahramani and Matthew J Beal. “Variational inference for Bayesian mixtures of factor analysers.” In: *Advances in neural information processing systems.* 2000, pp. 449–455.

[206] Thomas L Griffiths and Zoubin Ghahramani. “The indian buffet process: An introduction and review.” In: *Journal of Machine Learning Research* 12.Apr (2011), pp. 1185–1224.

[207] Adam Farooq. “A latent feature probabilistic principle component analysis model.” In: *Mathematics Project. Aston University.* 2018.

[208] Nils Lid Hjort et al. *Bayesian nonparametrics.* Vol. 28. Cambridge University Press, 2010.

[209] Ke Jiang, Brian Kulis, and Michael I Jordan. “Small-variance asymptotics for exponential family Dirichlet process mixture models.” In: *Advances in Neural Information Processing Systems.* 2012, pp. 3158–3166.

[210] Michael C Hughes and Erik Sudderth. “Memoized online variational inference for Dirichlet process mixture models.” In: *Advances in Neural Information Processing Systems.* 2013, pp. 1133–1141.

[211] Kenneth L Lange, Roderick JA Little, and Jeremy MG Taylor. “Robust statistical modeling using the t distribution.” In: *Journal of the American Statistical Association* 84.408 (1989), pp. 881–896.

[212] Sahibsingh A Dudani. “The distance-weighted k-nearest-neighbor rule.” In: *IEEE Transactions on Systems, Man, and Cybernetics* 4 (1976), pp. 325–327.

[213] Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. “A survey of dimensionality reduction techniques.” In: *arXiv preprint arXiv:1403.2877* (2014).

[214] Lei Chen. “Curse of Dimensionality.” In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 545–546. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_133. URL: https://doi.org/10.1007/978-0-387-39940-9_133.

[215] Yordan P Raykov et al. “What to do when k-means clustering fails: A simple yet principled alternative algorithm.” In: *PloS one* 11.9 (2016), e0162259.

[216] Yee Whye Teh. “Dirichlet process.” In: *Encyclopedia of machine learning* (2010), pp. 280–287.

[217] Neil D Lawrence. “Gaussian process latent variable models for visualisation of high dimensional data.” In: *Advances in neural information processing systems*. 2004, pp. 329–336.

[218] Yining Wang and Jun Zhu. “DP-space: Bayesian nonparametric subspace clustering with small-variance asymptotics.” In: *International Conference on Machine Learning*. 2015, pp. 862–870.

[219] Andreas E Kyprianou. *Introductory lectures on fluctuations of Lévy processes with applications*. Springer Science & Business Media, 2006.

[220] Yee Whye Teh. “Bayesian Nonparametrics Rough Notes.” In: (2012).

[221] Joshua B Tenenbaum, Vin De Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction.” In: *science* 290.5500 (2000), pp. 2319–2323.

[222] Peter Orbanz. “Lecture Notes on Bayesian Nonparametrics.” In: ().

[223] Parul Agarwal, M Afshar Alam, and Ranjit Biswas. "Issues, challenges and tools of clustering algorithms." In: *arXiv preprint arXiv:1110.2610* (2011).

[224] Li Qu et al. "PPCA-based missing data imputation for traffic flow volume: A systematical approach." In: *IEEE Transactions on intelligent transportation systems* 10.3 (2009), pp. 512–522.

[225] Geoffrey E Hinton and Sam T Roweis. "Stochastic neighbor embedding." In: *Advances in neural information processing systems*. 2003, pp. 857–864.

[226] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. "GTM: The generative topographic mapping." In: *Neural computation* 10.1 (1998), pp. 215–234.

[227] John Kingman. "Completely random measures." In: *Pacific Journal of Mathematics* 21.1 (1967), pp. 59–78.

[228] David S Broomhead and David Lowe. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Tech. rep. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[229] Alexander Khintchine. "Korrelationstheorie der stationären stochastischen Prozesse." In: *Mathematische Annalen* 109.1 (1934), pp. 604–615.

[230] JL Doob et al. "P. Lévy, Théorie de l'Addition des Variables Aléatoires." In: *Bulletin of the American Mathematical Society* 44.1, Part 1 (1938), pp. 19–20.

[231] Bruno de Finetti. "Integrazione delle funzioni ad incremento aleatorio." In: *Rendiconti della R. Accademia Nazionale dei Lincei* (1929), pp. 548–553.

[232] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. Vol. 80. Siam, 2002.

[233] *Brilliant Math & Science Wiki: Subspace*. Jan. 2019.

[234] *Peter Kempthorne, Choongbum Lee, Vasily Strela, and Jake Xia. 18.S096 Topics in Mathematics with Applications in Finance. Fall 2013. Massachusetts Institute of Technology: MIT OpenCourseWare*. Sept. 2013.

[235] RD Clarke. “An application of the Poisson distribution.” In: *Journal of the Institute of Actuaries* 72.3 (1946), pp. 481–481.

[236] Mark A Beaumont, Wenyang Zhang, and David J Balding. “Approximate Bayesian computation in population genetics.” In: *Genetics* 162.4 (2002), pp. 2025–2035.

[237] Charita Dellaporta et al. “Robust Bayesian Inference for Simulator-based Models via the MMD Posterior Bootstrap.” In: *arXiv preprint arXiv:2202.04744* (2022).

[238] SP Lyddon, CC Holmes, and SG Walker. “General Bayesian updating and the loss-likelihood bootstrap.” In: *Biometrika* 106.2 (2019), pp. 465–478.

[239] Krikamol Muandet et al. “Kernel mean embedding of distributions: A review and beyond.” In: *arXiv preprint arXiv:1605.09522* (2016).

[240] Max Welling and Kenichi Kurihara. “Bayesian K-means as a “maximization-expectation” algorithm.” In: *Proceedings of the 2006 SIAM international conference on data mining*. SIAM. 2006, pp. 474–478.

[241] Richard Royall and Tsung-Shan Tsou. “Interpreting statistical evidence by using imperfect models: robust adjusted likelihood functions.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.2 (2003), pp. 391–404.

[242] Wenxin Jiang and Martin A Tanner. “Gibbs posterior for variable selection in high-dimensional classification and data mining.” In: *The Annals of Statistics* 36.5 (2008), pp. 2207–2231.

[243] Tong Zhang. “From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation.” In: *The Annals of Statistics* 34.5 (2006), pp. 2180–2210.

[244] CC Holmes and SG Walker. “Assigning a value to a power likelihood in a general Bayesian model.” In: *Biometrika* 104.2 (2017), pp. 497–503.

[245] Ajay Jasra, Chris C Holmes, and David A Stephens. “Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling.” In: *Statistical Science* (2005), pp. 50–67.

[246] Thomas L Griffiths and Zoubin Ghahramani. "Infinite latent feature models and the Indian buffet process." In: *NIPS*. Vol. 18. 2005, pp. 475–482.

[247] Diana Cai, Trevor Campbell, and Tamara Broderick. "Finite mixture models do not reliably learn the number of components." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 1158–1169.

[248] David A Binder. "Bayesian cluster analysis." In: *Biometrika* 65.1 (1978), pp. 31–38.

[249] Klaus Th Hess, Anett Liewald, and Klaus D Schmidt. "An extension of Panjer's recursion." In: *ASTIN Bulletin: The Journal of the IAA* 32.2 (2002), pp. 283–297.

[250] Michael Fackler and DAV Aktuar. "Panjer class united–one formula for the Poisson, Binomial, and Negative Binomial distribution." In: *ASTIN colloquium*. 2009.

[251] Harry H Panjer. "Recursive evaluation of a family of compound distributions." In: *ASTIN Bulletin: The Journal of the IAA* 12.1 (1981), pp. 22–26.

[252] Peter Orbanz and Yee Whye Teh. "Bayesian Nonparametric Models." In: *Encyclopedia of machine learning* 1 (2010).

[253] CC Holmes and SG Walker. "Assigning a value to a power likelihood in a general Bayesian model." In: *Biometrika* 104.2 (2017), pp. 497–503.

[254] Nial Friel and Anthony N Pettitt. "Marginal likelihood estimation via power posteriors." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.3 (2008), pp. 589–607.

[255] Charles J Geyer. "Markov chain Monte Carlo maximum likelihood." In: (1991).

[256] Stephen Walker and Nils Lid Hjort. "On bayesian consistency." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.4 (2001), pp. 811–821.

[257] Yee Whye Teh et al. "Hierarchical dirichlet processes." In: *Journal of the american statistical association* 101.476 (2006), pp. 1566–1581.

[258] David J Aldous. "Exchangeability and related topics." In: *École d'Été de Probabilités de Saint-Flour XIII—1983*. Springer, 1985, pp. 1–198.

[259] Yee Whye Teh. *Dirichlet Process.* 2010.

[260] Zoubin Ghahramani and Thomas L Griffiths. “Infinite latent feature models and the Indian buffet process.” In: *Advances in neural information processing systems.* 2006, pp. 475–482.

Appendix A

Generalized count mixture models supplementary material

A.1 Panjer distribution generalisation

This section will show how the Panjer distribution is a generalisation of the binomial, Poisson and negative-binomial distribution.

Binomial distribution

The PMF of the Panjer distribution (from equation (3.6)) can be re-written as:

$$\begin{aligned}
\mathrm{P}\left(y|\lambda,\eta\right) &= \left(1+\frac{\lambda}{\eta}\right)^{-\eta}\frac{\lambda^y}{y!}\prod_{i=0}^{y-1}\frac{\eta+i}{\eta+\lambda} \\
&= \frac{\lambda^y}{y!}\left(-1\right)^y\frac{\eta^\eta}{\left(\eta+\lambda\right)^{\eta+y}}\left(-1\right)^y\prod_{i=0}^{y-1}\left(\eta+i\right) \\
&= \frac{1}{y!}\frac{\lambda^y\left(-\eta-\lambda\right)^{-\eta-y}}{-\eta^{-\eta}}\prod_{i=0}^{y-1}\left(-\eta-i\right),
\end{aligned}$$

if $m = -\eta$:

$$\begin{aligned}
\mathrm{P}\left(y|\lambda, m\right) &= \binom{m}{y} \frac{\lambda^y \left(m-\lambda\right)^{m-y}}{m^m} \\
&= \binom{m}{y} \frac{\lambda^y \left(m-\lambda\right)^{m}}{\left(m-\lambda\right)^y m^m} \\
&= \binom{m}{y} \left(\frac{\lambda}{m-\lambda}\right)^y \left(\frac{m}{m-\lambda}\right)^{-m} \\
&= \binom{m}{y} \left(\frac{\lambda}{m}\right)^y \left(\frac{m}{m-\lambda}\right)^y \left(\frac{m}{m-\lambda}\right)^{-m} \\
&= \binom{m}{y} \left(\frac{\lambda}{m}\right)^y \left(\frac{m-\lambda}{m}\right)^{m-y} \\
&= \binom{m}{y} \left(p\right)^y \left(1-p\right)^{m-y},
\end{aligned}$$

which is the binomial distribution with m number of trails and probability of success p.

Negative binomial distribution

The different terms of the Panjer PMF (from equation (3.6)) can be re-written as:

$$\frac{1}{y!}\prod_{i=0}^{y-1}\left(\eta+i\right) = \binom{\eta+y-1}{y},$$

$$\left(1+\frac{\lambda}{\eta}\right)^{-\eta} = \left(\frac{\eta}{\eta+\lambda}\right)^{\eta},$$

$$\lambda^y \prod_{i=0}^{y-1} \frac{1}{\eta+\lambda} = \left(\frac{\lambda}{\eta+\lambda}\right)^{y},$$

if $r = \eta$:

$$\begin{aligned}
\mathrm{P}\left(y|r, \lambda\right) &= \binom{r+y-1}{y} \left(\frac{r}{r+\lambda}\right)^r \left(\frac{\lambda}{r+\lambda}\right)^y \\
&= \binom{r+y-1}{y} \left(1-p\right)^r \left(p\right)^y,
\end{aligned}$$

which is the negative binomial distribution with probability of success $p = \frac{\lambda}{r+\lambda}$ and r number of failures allowed.

Poisson distribution

By taking the limit $\eta \to \infty$, the PMF of the Panjer distribution (from equation (3.6)) can be re-written as:

$$\begin{aligned}
\lim_{\eta\to\infty} \mathrm{P}\left(y|\lambda\right) &= \lim_{\eta\to\infty} \left(\left(1 + \frac{\lambda}{\eta}\right)^{-\eta} \frac{\lambda^y}{y!} \prod_{i=0}^{y-1} \frac{\eta + i}{\eta + \lambda} \right) \\
&= \frac{\lambda^y}{y!} \lim_{\eta\to\infty} \left(\left(1 + \frac{\lambda}{\eta}\right)^{-\eta} \prod_{i=0}^{k-1} \frac{\eta + i}{\eta + \lambda} \right) \\
&= \frac{\lambda^y}{y!} \lim_{\eta\to\infty} \left(\left(\left(1 + \frac{\lambda}{\eta}\right)^{\eta/\lambda} \right)^{-\lambda} \prod_{i=0}^{y-1} \frac{\eta + i}{\eta + \lambda} \right) \\
&= \frac{\lambda^y}{y!} \exp\left(-\lambda\right),
\end{aligned}$$

which gives the Poisson distribution with rate parameter λ.

A.2 Exponential family of distribution

This section shows how the Panjer distribution cannot be expressed in the exponential family form which (slightly different notation from equation (2.2)) takes the form:

$$\mathrm{P}\left(y|\theta\right) = \exp\left(\gamma\left(\theta\right) T\left(y\right) - A\left(\gamma\left(\theta\right)\right) + C\left(y\right)\right),$$

where θ is the distributional parameter, $\gamma(\cdot)$ is the natural parameter, $T(\cdot)$ is the sufficient statistic, $A\left(\cdot\right)$ is the log-partition and $C\left(\cdot\right)$ is the log-base measure. The log of which takes the form:

$$\ln \mathrm{P}\left(y|\theta\right) = \gamma\left(\theta\right) T\left(y\right) - A\left(\gamma\left(\theta\right)\right) + C\left(y\right).$$

The same PMF from equation (3.6) will be used throughout this section; however, the derivations will be done using two cases; each of which has different assumptions on the η parameter.

Case 1

Let $\eta > 0$, the log of equation (3.6) is:

$$\ln \mathrm{P}\left(y|\lambda,\eta\right) = y\ln\left(\frac{\lambda}{\eta+\lambda}\right) - \eta\ln\left(1+\frac{\lambda}{\eta}\right) - \ln\left(y!\right) + \sum_{i=0}^{y-1}\ln\left(\eta+i\right),$$

which cannot be rewritten in the log exponential family form due to the coupling of the parameters λ and η. From this, it can be concluded that the Panjer distribution with $\eta > 0$ cannot be expressed in the exponential family form.

Case 2

Let $\eta < -\lambda$, then equation (3.6) can be re-written as:

$$\mathrm{P}\left(y|\lambda,\eta\right) = \frac{\lambda^y\left(-\eta-\lambda\right)^{-\eta-y}}{-\eta^{-\eta}}\prod_{i=0}^{y-1}\left(-\eta-i\right)\frac{1}{y!},$$

the log of the above results in:

$$\begin{aligned}\ln \mathrm{P}\left(y|\lambda,\eta\right) &= y\ln\left(\lambda\right) - y\ln\left(-\eta-\lambda\right) - \eta\ln\left(-\eta-\lambda\right) + \eta\ln\left(-\eta\right)\\ &\qquad + \left[\sum_{i=0}^{y-1}\ln\left(-\eta-i\right)\right] - \ln\left(y!\right)\\ &= y\ln\left(\frac{\lambda}{-\eta-\lambda}\right) + \eta\ln\left(\frac{-\eta}{-\eta-\lambda}\right) + \left[\sum_{i=0}^{y-1}\ln\left(-\eta-i\right)\right] - \ln\left(y!\right)\\ &= y\ln\left(\frac{\lambda}{-\eta-\lambda}\right) - \eta\ln\left(1-\frac{\lambda}{-\eta}\right) + \left[\sum_{i=0}^{y-1}\ln\left(-\eta-i\right)\right] - \ln\left(y!\right),\end{aligned}$$

which cannot be re-written in the log exponential family form due to the coupling of the parameters λ and η. From this, it can be concluded that the Panjer distribution with $\eta < -\lambda$ cannot be expressed in the exponential family form.

A.3 Maximum likelihood solution

This section will derive the maximum likelihood updates for the parameters of the Panjer distribution. Throughout this section, it will be assumed that $\{y_n\}_{n=1}^N$ are some N samples from the Panjer distribution parameterised by λ and η (see equation (3.6)); the data likelihood of which is:

$$\mathrm{P}\left(\{y_n\}_{n=1}^N|\eta,\lambda\right)=\left(1+\frac{\lambda}{\eta}\right)^{-N\eta}\prod_{n=1}^N\left(\frac{\lambda^{y_n}}{y_n!}\left[\prod_{i=0}^{y_n-1}\frac{\eta+i}{\eta+\lambda}\right]\right).$$

A.3.1 Maximum likelihood solution for λ

The log of the data likelihood is:

$$\begin{aligned}\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^N|\eta,\lambda\right)\right)&\propto -N\eta\ln\left(1+\frac{\lambda}{\eta}\right)+\sum_{n=1}^N\left[y_n\ln\left(\lambda\right)-y_n\ln\left(\eta+\lambda\right)\right]\\&\propto -N\eta\ln\left(\eta+\lambda\right)+\sum_{n=1}^N\left[y_n\ln\left(\lambda\right)-y_n\ln\left(\eta+\lambda\right)\right],\end{aligned}$$

where terms that don't depend on λ have been removed, the derivative with respect to λ is:

$$\frac{d\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^N|\eta,\lambda\right)\right)}{d\lambda}=\frac{-N\eta}{\eta+\lambda}-\frac{\sum_{n=1}^N y_n}{\eta+\lambda}+\frac{\sum_{n=1}^N y_n}{\lambda}.$$

By setting the derivative to zero:

$$\begin{aligned}\frac{\sum_{n=1}^N y_n}{\lambda}&=\frac{N\eta+\sum_{n=1}^N y_n}{\eta+\lambda},\\\frac{\eta+\lambda}{\lambda}&=\frac{N\eta+\sum_{n=1}^N y_n}{\sum_{n=1}^N y_n},\\\frac{\eta}{\lambda}&=\frac{N\eta+\sum_{n=1}^N y_n}{\sum_{n=1}^N y_n}-1\\&=\frac{N\eta}{\sum_{n=1}^N y_n},\end{aligned}$$

which results in the following maximum likelihood update for λ:

$$\lambda_{\mathrm{ML}} = \frac{\sum_{n=1}^{N} y_n}{N}.$$

A.3.2 Maximum likelihood solution for η

The maximum likelihood solution update for η will be derived using two cases, each of which will have a different assumption on the η parameter.

Case 1

Let $\eta > 0$, then the data log-likelihood is:

$$\begin{aligned}\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\eta,\lambda\right)\right) \propto &- N\eta\left[\ln\left(\eta+\lambda\right) - \ln\left(\eta\right)\right] - \sum_{n=1}^{N}\left[y_n \ln\left(\eta+\lambda\right)\right] \\ &+ \sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\ln\left(\eta+i\right)\right],\end{aligned}$$

where terms that don't depend on η have been removed, the derivative with respect to η is:

$$\begin{aligned}\frac{d\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\eta,\lambda\right)\right)}{d\eta} &= \frac{N\lambda}{\eta+\lambda} - N\left(\ln\left(\eta+\lambda\right) - \ln\left(\eta\right)\right) - \frac{\sum_{n=1}^{N} y_n}{\eta+\lambda} \\ &\quad + \sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right] \\ &= \frac{N\lambda - \sum_{n=1}^{N} y_n}{\eta+\lambda} - N\left(\ln\left(\eta+\lambda\right) - \ln\left(\eta\right)\right) + \sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right],\end{aligned}$$

by using $\lambda = \frac{\sum_{n=1}^{N} y_n}{N}$ the above can be simplified to:

$$\frac{d\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\eta,\lambda\right)\right)}{d\eta} = -N\left(\ln\left(\eta+\lambda\right) - \ln\left(\eta\right)\right) + \sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right],$$

which has no closed-form solution when set to zero. The second derivative of the data log-likelihood with respect to η can be written as:

$$\frac{d^2 \ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\eta,\lambda\right)\right)}{d\eta^2} = \frac{N\lambda}{\eta\left(\eta+\lambda\right)} - \sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\left(\eta+i\right)^2}\right].$$

Case 2

Let $\eta < -\lambda$, then equation (3.6) can be re-written as:

$$\mathrm{P}\left(y|\lambda,\eta\right) = \frac{\lambda^{y}\left(-\eta-\lambda\right)^{-\eta-y}}{-\eta^{-\eta}}\prod_{i=0}^{y-1}\left(-\eta-i\right)\frac{1}{y!},$$

which would result in the following data likelihood:

$$\begin{aligned}\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\lambda,\eta\right) &= \prod_{n=1}^{N}\frac{\lambda^{y_n}\left(-\eta-\lambda\right)^{-\eta-y_n}}{-\eta^{-\eta}}\frac{1}{y_n!}\left[\prod_{i=0}^{y_1-1}\left(-\eta-i\right)\right]\\ &= \frac{\lambda^{\sum_{n=1}^{N}y_n}\left(-\eta-\lambda\right)^{-N\eta-\sum_{n=1}^{N}y_n}}{-\eta^{-N\eta}}\prod_{n=1}^{N}\left[\frac{1}{y_n!}\prod_{i=0}^{y_1-1}\left(-\eta-i\right)\right],\end{aligned}$$

the log of which is:

$$\begin{aligned}\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\lambda,\eta\right)\right) \propto &-\left(\sum_{n=1}^{N}y_n+N\eta\right)\ln\left(-\eta-\lambda\right)+N\eta\ln\left(-\eta\right)\\ &+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\ln\left(-\eta-i\right)\right],\end{aligned}$$

where terms that don't depend on η have been removed, the derivative with respect to η is:

$$\begin{aligned}\frac{d\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\lambda,\eta\right)\right)}{d\eta} =&-\frac{-N\eta-\sum_{n=1}^{N}y_n}{-\lambda-\eta}-N\left(\ln\left(-\lambda-\eta\right)\right)+N\ln\left(-\eta\right)+N\\&+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right]\\=&-N\left(\ln\left(-\lambda-\eta\right)-\ln\left(-\eta\right)\right)-\frac{-N\eta-\sum_{n=1}^{N}y_n}{-\lambda-\eta}\\&+\frac{N\left(-\lambda-\eta\right)}{-\lambda-\eta}+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right],\end{aligned}$$

by using $\lambda=\frac{\sum_{n=1}^{N}y_n}{N}$, the above can be simplified to:

$$\begin{aligned}\frac{d\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\lambda,\eta\right)\right)}{d\eta}=&-N\left(\ln\left(-\lambda-\eta\right)-\ln\left(-\eta\right)\right)\\&+\frac{N\eta+\sum_{n=1}^{N}y_n-N\lambda-N\eta}{-\lambda-\eta}\\&+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right]\\=&-N\left(\ln\left(-\lambda-\eta\right)-\ln\left(-\eta\right)\right)\\&+\frac{N\eta+\sum_{n=1}^{N}y_n-N\frac{1}{N}\left(\sum_{n=1}^{N}y_n\right)-N\eta}{-\lambda-\eta}\\&+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right]\\=&-N\left(\ln\left(-\lambda-\eta\right)-\ln\left(-\eta\right)\right)+\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\eta+i}\right],\end{aligned}$$

which has no closed-form solution when set to zero. The second derivative of the data log-likelihood with respect to η can be written as:

$$\frac{d^2\ln\left(\mathrm{P}\left(\{y_n\}_{n=1}^{N}|\lambda,\eta\right)\right)}{d\eta^2}=\frac{N\lambda}{\eta\left(\eta+\lambda\right)}-\sum_{n=1}^{N}\left[\sum_{i=0}^{y_n-1}\frac{1}{\left(\eta+i\right)^2}\right].$$

Combining case 1 and case 2

By combining both cases, the first and second derivative of the Panjer distributing with respect to η can be written in the form:

$$\frac{d \ln \left(\mathrm{P}\left(\{y_n\}_{n=1}^{N} | \lambda, \eta \right) \right)}{d\eta} = -N \left(\ln \left(\operatorname{sgn}\left(\eta\right) \left(\lambda + \eta\right) \right) - \ln \left(\operatorname{sgn}\left(\eta\right) \eta \right) \right) + \sum_{n=1}^{N} \left[\sum_{i=0}^{y_n - 1} \frac{1}{\eta + i} \right],$$

$$\frac{d^2 \ln \left(\mathrm{P}\left(\{y_n\}_{n=1}^{N} | \lambda, \eta \right) \right)}{d\eta^2} = \frac{N\lambda}{\eta \left(\eta + \lambda\right)} - \sum_{n=1}^{N} \left[\sum_{i=0}^{y_n - 1} \frac{1}{\left(\eta + i\right)^2} \right],$$

where $\operatorname{sgn}\left(\cdot\right)$ is the sign function, such that:

$$\operatorname{sgn}\left(x\right) = \begin{cases} -1 & \text{if} \quad x < 1 \\ 0 & \text{if} \quad x = 0 \cdot \\ 1 & \text{if} \quad x > 1 \end{cases}$$

A.4 Inference

This section highlights the two different schemes of learning the parameters of the Panjer mixture model. A D-dimensional generalisation of the Panjer PMF (from equation (3.6)) is a product of D number of one-dimensional (univariate) Panjer distributions, such that:

$$\mathrm{P}\left(\mathbf{y} | \boldsymbol{\lambda}, \boldsymbol{\eta}\right) = \prod_{d=1}^{D} \mathrm{P}\left(y_d | \lambda_d, \eta_d\right),$$

where $\mathbf{y}$ is some D-dimensional observation and $\{\boldsymbol{\lambda}, \boldsymbol{\eta}\}$ are D-dimensional parameters; note that $y_d \in \mathbb{N}_0$, $\lambda_d > 0$ and $\eta_d \in \{[-\infty, -\lambda_d) \cup (0, \infty]\}$. Then under a K component D-dimensional Panjer mixture model, the probability of some D-dimensional observation $\mathbf{y}_n$ and its respective latent variable $\mathbf{z}_n$ (see Section 3.3.1) given all parameters is:

$$\mathrm{P}\left(\mathbf{y}_n, \mathbf{z}_n | \boldsymbol{\theta}, \boldsymbol{\pi}\right) = \prod_{k=1}^{K} \left(\pi_k \prod_{d=1}^{D} \mathrm{P}\left(y_{nd} | \lambda_{kd}, \eta_{kd}\right) \right)^{z_{nk}},$$

where $\boldsymbol{\theta} = \{\boldsymbol{\lambda}_k, \boldsymbol{\eta}_k\}_{k=1}^{K}$ contains all the parameters, and $\pi_k = \mathrm{P}\left(z_{nk} = 1\right)$. Then after N observations the complete data likelihood is:

$$\mathrm{P}\left(\{\mathbf{y}_n, \mathbf{z}_n\}_{n=1}^{N} | \boldsymbol{\theta}, \boldsymbol{\pi}\right) = \prod_{n=1}^{N}\prod_{k=1}^{K}\left(\pi_k \prod_{d=1}^{D} \mathrm{P}\left(y_{nd}|\lambda_{kd}, \eta_{kd}\right)\right)^{z_{nk}},$$

which has the log-likelihood:

$$\mathcal{L}_N = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\Bigg[\ln \pi_k + \sum_{d=1}^{D}\Bigg\{ - \eta_{kd} \ln\left(1 + \frac{\lambda_{kd}}{\mu_{kd}}\right) + y_{nd} \ln\left(\lambda_{kd}\right)$$
$$- \ln\left(y_{nd}!\right) + \sum_{i=0}^{y_n - 1}\left(\ln\left(\frac{\eta_{kd} + i}{\eta_{kd} + \lambda_{kd}}\right)\right)\Bigg\}\Bigg].$$

A.4.1 Expectation-maximisation

The first scheme to infer all the parameters is the expectation-maximisation (EM) algorithm [45]. The first step is to take the expectation (E-step) of the latent parameters z_{nk} with respect to the posterior distribution, this is evaluated as:

$$\gamma\left(z_{nk}\right) = \frac{\pi_k \mathrm{P}\left(\mathbf{y}_n|\boldsymbol{\lambda}_k, \boldsymbol{\eta}_k\right)}{\sum_{j=1}^{K} \pi_j \mathrm{P}\left(\mathbf{y}_n|\boldsymbol{\lambda}_j, \boldsymbol{\eta}_j\right)},$$

which is done for all $n = \{1, \ldots, N\}$ and $k = \{1, \ldots, K\}$. This is directly followed by the second step which maximises (M-step) the complete data log-likelihood with respect to the other parameters. A Lagrange multiplier is added onto the complete data log-likelihood to ensure $\sum_k \pi_k$. The M-step updates are:

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N}\gamma\left(z_{nk}\right),$$

which is done for all $k = \{1, \ldots, K\}$.

$$\lambda_{kd} = \pi_k \sum_{n=1}^{N}\gamma\left(z_{nk}\right) y_{nd},$$

which is done for all $k = \{1, \ldots, K\}$ and $d = \{1, \ldots, D\}$. The parameter η_{kd} has no closed-form solution (see Appendix A.3.2) and is therefore learned numerically using the Newton-Raphson method initialised with:

$$\eta^{*}_{kd} = \frac{\lambda^{2}_{kd}}{\frac{\sum_{n=1}^{N} \gamma(z_{nk})(y_{nd} - \lambda_{kd})^{2}}{\sum_{n=1}^{N} \gamma(z_{nk}) - 1} - \lambda_{kd}},$$

which is motivated by the method of moments, this is done for all $k = \{1, \ldots, N\}$ and $d = \{1, \ldots, D\}$. This scheme iterative applies the E step followed by the M step till the difference in the complete data log-likelihood is less than some pre-defined threshold.

A.4.2 Maximisation-maximisation

The second scheme to infer all the parameters is the maximisation-maximisation (MM) algorithm. The first step is to maximise (M-step) the complete data log-likelihood with respect to the latent parameters z_{nk}, this is done by first finding:

$$k = \operatorname{argmax} \left(\pi_1 \mathrm{P}\left(\mathbf{y}_n | \boldsymbol{\lambda}_1, \boldsymbol{\eta}_1\right), \ldots, \pi_K \mathrm{P}\left(\mathbf{y}_n | \boldsymbol{\lambda}_K, \boldsymbol{\eta}_K\right) \right),$$

and then setting $z_{nk} = 1$ and $z_{ni} = 0$ for $i \neq k$; this is done for all $n = \{1, \ldots, N\}$. This is directly followed by the second step which maximises (M-step) the complete data log-likelihood with respect to the other parameters. A Lagrange multiplier is added onto the complete data log-likelihood to ensure $\sum_k \pi_k = 1$. The M-step updates are:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{n} z_{nk},$$

which is done for all $k = 1, \ldots, K$.

$$\lambda_{kd} = \pi_k \sum_{n=1}^{N} \gamma\left(z_{nk}\right) y_{nd},$$

which is done for all $k = \{1, \ldots, K\}$, and $d = \{1, \ldots, D\}$. The parameter η_{kd} has no closed-form solution (see Appendix A.3.2) and is therefore learnt numerically using the

Newton-Raphson method initialised with:

$$\eta^*_{kd} = \frac{\lambda^2_{kd}}{\frac{\sum_{n=1}^{N} \gamma(z_{nk})(y_{nd} - \lambda_{kd})^2}{\sum_{n=1}^{N} \gamma(z_{nk}) - 1} - \lambda_{kd}},$$

which is motivated by the method of moments, this is done for all $k = \{1, \ldots, N\}$ and $d = \{1, \ldots, D\}$. This scheme iterative applies the E-step followed by the M-step till the difference in the complete data log-likelihood is less than some pre-defined threshold.

Appendix B

Robustifying mixture models using integral probability metric: supplementary material

B.1 Inference

In this section, we propose an expectation-maximisation (EM) scheme to infer all the parameters in the neighborhood mixture model from Section 4. The parameters $\{c_n\}_{n=1}^N$ are replaced by latent variables $\{\mathbf{z}_n\}_{n=1}^N$ indicate which mixture an observation was sampled from, such that if $z_{nk} = 1$ then observation n was sampled from cluster k; this is done for notational convenience.

The data likelihood of observing N observations is:

$$\mathrm{P}\left(\{\mathbf{y}_n, \mathbf{z}_n\}_{n=1}^N | \{\mathbf{u}_m^{(1)}\}_{m=1}^M, \ldots, \{\mathbf{u}_m^{(K)}\}_{m=1}^M, \boldsymbol{\pi}\right) = \prod_{n=1}^N \prod_{k=1}^K \left(\pi_k \mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m^{(k)}\}_{m=1}^M\right)\right)^{z_{nk}},$$

where $\mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m^{(k)}\}_{m=1}^M\right)$ is defined in equation (4.5), the log of which gives the following data log-likelihood:

$$\mathcal{L}_N = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left(\ln \pi_k + \ln \left(\mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m^{(k)}\}_{m=1}^M\right)\right)\right).$$

We then take the expectation (E-step) of the log-likelihood with respect to the latent variables z_{nk}; this results in the following update:

$$\gamma\left(z_{nk}\right) = \frac{\pi_k \mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m^{(k)}\}_{m=1}^{M}\right)}{\sum_{j=1}^{K} \pi_j \mathrm{P}\left(\mathbf{y}_n | \{\mathbf{u}_m^{(j)}\}_{m=1}^{M}\right)},$$

which is done for all $n = \{1, \dots, N\}$ and $k = \{1, \dots, K\}$. This is directly followed by the third step which maximises (M-step) the complete data log-likelihood with respect to the other parameters. A Lagrange multiplier is added onto the complete data log-likelihood to ensure $\sum_{k=1}^{K} \pi_k$, this results in the following update:

$$\tilde{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \gamma\left(z_{nk}\right),$$

which is done for all $k = \{1, \dots, K\}$. Then for each $k = \{1, \dots, K\}$ the complete data log-likelihood is maximised with respect to the 'pseudo-points', which has no closed form update, therefore, for all $m = \{1, \dots, M\}$, and $k = \{1, \dots, K\}$ it is updated iteratively using gradient decent:

$$\begin{aligned}\tilde{\mathbf{u}}_m^{(k)} &= \\ &\tilde{\mathbf{u}}_m^{(k)} - \alpha \frac{d}{d\mathbf{u}_m^{(k)}} \left(\lambda \left(\frac{\tilde{\pi}_k}{M} \sum_{i=1}^{M} k\left(\mathbf{u}_m^{(k)}, \mathbf{u}_i^{(k)}\right) - 2 \sum_{n=1}^{N} \gamma\left(z_{nk}\right) k\left(\mathbf{u}_m^{(k)}, \mathbf{y}_n\right) \right) \right),\end{aligned} \tag{B.1}$$

where $\alpha > 0$ is a step constant. If the kernel $k\left(\cdot, \cdot\right)$ is a Gaussian, then:

$$k\left(\mathbf{x}, \mathbf{y}\right) = \exp\left(-\frac{1}{\sigma^2} \left\| \mathbf{x} - \mathbf{y} \right\| \right),$$

then equation (B.1) becomes:

$$\tilde{\mathbf{u}}_m^{(k)} = \tilde{\mathbf{u}}_m^{(k)} - \alpha \left(\lambda \left(\frac{\tilde{\pi}_k}{M} \sum_{i=1}^{M} \tilde{k}\left(\mathbf{u}_m^{(k)}, \mathbf{u}_i^{(k)}\right) - 2 \sum_{n=1}^{N} \gamma\left(z_{nk}\right) \tilde{k}\left(\mathbf{u}_m^{(k)}, \mathbf{y}_n\right) \right) \right),$$

where:

$$\tilde{k}\left(\mathbf{x}, \mathbf{y}\right) = -\frac{2}{\sigma^2} \exp\left(-\frac{1}{\sigma^2} \left\| \mathbf{x} - \mathbf{y} \right\| \right) \times \left(\mathbf{x} - \mathbf{y}\right).$$

B.2 Results

In this section we present the results of all experiments from the data set presented in Figure 4.2.

	ζ value		
	$\zeta = 0.1$	$\zeta = 0.5$	$\zeta = 0.9$
Train Data (EM)	**85 ± 2**	79 ± 1	76 ± 1
Train Data (Gibbs)	**84 ± 3**	75 ± 2	72 ± 1
Test Data (EM)	**81 ± 1**	68 ± 3	66 ± 2
Test Data (Gibbs)	**81 ± 2**	74 ± 3	73 ± 1

Table B.1: Performance of likelihood tempering Gaussian mixture models with different ζ values on synthetic data set in Figure 4.2. Each model was trained on 80% of the data set and tested on the remaining 20%. The average accuracy and one standard deviation from 20 different experiments are reported.

		$B = 100$		$B = 1000$	
		$T = 100$	$T = 100$	$T = 100$	$T = 100$
$c = 1$	Train Data (Gibbs)	73 ± 1	74 ± 2	80 ± 2	79 ± 2
	Test Data (Gibbs)	72 ± 3	75 ± 2	76 ± 2	78 ± 1
$c = 10$	Train Data (Gibbs)	78 ± 1	80 ± 2	80 ± 1	**82 ± 1**
	Test Data (Gibbs)	78 ± 3	79 ± 1	79 ± 2	**80 ± 2**
$c = 100$	Train Data (Gibbs)	71 ± 1	74 ± 1	75 ± 1	75 ± 3
	Test Data (Gibbs)	71 ± 3	72 ± 2	74 ± 3	76 ± 3

Table B.2: Performance of likelihood posterior bootstrapping mixture models with different parameter values on synthetic data set from Figure 4.2. Each model was trained on 80% of the data set and tested on the remaining 20%. The average accuracy and one standard deviation from 20 different experiments are reported.

Appendix C

Piecewise linear dimensionality reduction clustering supplementary material

C.1 Latent variable PPCA relationship to PCA

In this section we demonstrate that the latent variable probablistic principal component analysis model from Section 5.6 is indeed a generalisation of the ubiquitous PCA and using *small variance asymptotics* [92]; this includes the infinite sparse PPCA and the adaptive PPCA. Let us first start by marginalising out the discrete and continuous latent variables $\{\mathbf{x}_n, \mathbf{z}_n\}$ which are not of explicit interest in conventional PCA approach. To compute the marginal likelihood of $\mathbf{y}_n$ we compute the expectations: $\mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}[\mathbf{y}_n]$, and $\mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}\left[(\mathbf{y}_n - \mathbb{E}[\mathbf{y}_n])(\mathbf{y}_n - \mathbb{E}[\mathbf{y}_n])^T\right]$, where we will use $\mathbb{E}[\cdot] = \mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}[\cdot]$ for notational convenience. We express the moments of the marginal likelihood starting with the posterior mean of the marginal:

$$\begin{aligned}\mathbb{E}\left[\mathbf{y}_n\right] &= \mathbb{E}\left[\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n\right] \\ &= \mathbf{W}\left(\mathbb{E}\left[\mathbf{x}_n\right] \odot \mathbb{E}\left[\mathbf{z}_n\right]\right) + \boldsymbol{\mu} + \mathbb{E}\left[\boldsymbol{\epsilon}_n\right] \\ &= \mathbf{W}\left(0 \odot \boldsymbol{\rho}\right) + \boldsymbol{\mu} + 0 \\ &= \boldsymbol{\mu},\end{aligned}$$

where we have used a diagonal $(K \times K)$ matrix $\boldsymbol{\rho}$ to denote the expectation of each feature, which is determined by the prior on the matrix $\mathbf{Z}$:

$$\rho_{kk} = \begin{cases} \frac{L}{K} & \text{if multivariate hypergeometric prior} \\ \frac{1}{N}\sum_{n=1}^{N} z_{kn} & \text{if IBP prior} \end{cases}.$$

For the variance of the marginal, we can write:

$$\begin{aligned}\mathbb{E}\left[\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)^T\right] &= \mathbb{E}\left[\left(\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n\right)\left(\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n\right)^T\right] \\ &= \mathbf{W}\boldsymbol{\rho}\mathbf{W}^T + \sigma^2\mathbf{I}_D,\end{aligned}$$

where $\mathbf{I}_D$ is an identity matrix of size D. Finally, using the obtained expression for $\mathbb{E}\left[\mathbf{y}_n\right]$ and $\mathbb{E}\left[\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)^T\right]$, combined with the Gaussian likelihood of $\mathbf{y}_n$ resulting in a linear Gaussian model, we can write the marginal likelihood as:

$$\mathrm{P}\left(\mathbf{y}_n \mid \mathbf{W}, \boldsymbol{\rho}, \sigma\right) = \frac{1}{\left(2\pi\right)^{\frac{D}{2}}}\left|\mathbf{C}\right|^{-1/2}\exp\left(-\frac{1}{2}\mathbf{y}_n^T\mathbf{C}^{-1}\mathbf{y}_n\right),$$

where we used $\mathbf{C} = \mathbf{W}\boldsymbol{\rho}\mathbf{W}^T + \sigma^2\mathbf{I}_D$ to denote the model covariance.

Now, the marginal likelihood in this collapsed latent variable PPCA model is almost identical to the PPCA model [184] with the key difference being the weights $\boldsymbol{\rho}$ which can be scalar shared across each dimension or direction specific. In fact, we can say that the PPCA model is a special case of the collapsed latent variable PPCA model when the diagonal of $\boldsymbol{\rho}$ are full of ones, which occurs when the matrix $\mathbf{Z}$ is full of ones implying all observations are active in all K number of one-dimensional subspaces.

The complete data log-likelihood of the collapsed model is:

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \ln\left(\mathrm{P}\left(\mathbf{y}_n \mid \mathbf{W}, \boldsymbol{\rho}, \sigma\right)\right) \\ &= -\frac{N}{2}\left(D \ln(2\pi) + \ln|\mathrm{C}| + \mathrm{tr}\left(\mathrm{C}^{-1}\mathrm{S}\right)\right)\end{aligned}$$

where $\mathrm{tr}(\cdot)$ is a trace function, and $\mathrm{S} = \frac{1}{N}\mathrm{YY}^T$. To find the maximum likelihood estimates for $\mathbf{W}$, we differentiate the likelihood and solve:

$$\frac{d\mathcal{L}}{d\mathbf{W}} = -\frac{N}{2}\left(2\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho} - 2\mathrm{C}^{-1}\mathrm{S}\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho}\right) = 0.$$

The maximum likelihood estimate for $\mathbf{W}$ then should satisfy:

$$\begin{aligned}\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho} &= \mathrm{C}^{-1}\mathrm{S}\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho}, \\ \mathbf{W}^{\mathrm{ML}}\boldsymbol{\rho} &= \mathrm{S}\mathrm{C}^{-1}\mathbf{W}^{\mathrm{ML}}\boldsymbol{\rho}.\end{aligned}$$

To find the solution for the above we first express the $\mathbf{W}\boldsymbol{\rho}^{1/2}$ term using its singular value decomposition:

$$\mathbf{W}\boldsymbol{\rho}^{1/2} = \mathbf{ULV}^T,$$

which leads to:

$$\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho}^{1/2} = \mathbf{UL}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)^{-1}\mathbf{V}^T,$$

then:

$$\begin{aligned}\mathrm{S}\mathrm{C}^{-1}\mathbf{W}\boldsymbol{\rho}^{1/2} &= \mathbf{W}\boldsymbol{\rho}^{1/2}, \\ \mathrm{S}\mathbf{UL}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)^{-1}\mathbf{V}^T &= \mathbf{ULV}^T, \\ \mathrm{S}\mathbf{UL} &= \mathbf{U}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)\mathbf{L},\end{aligned}$$

which implies that $\mathbf{u}_j$ is the eigenvector of S with eigenvalue of $\lambda_j = \sigma^2 + l_j^2$. Therefore all potential solutions for $\mathbf{W}^{\mathrm{ML}}$ may be written as:

$$\mathbf{W}^{\mathrm{ML}} = \mathbf{U}_K \left(\mathbf{K}_K - \sigma^2 \mathbf{I}_K\right)^{1/2} \mathbf{R}\boldsymbol{\rho}^{-1/2},$$

where:

$$k_{jj} = \begin{cases} \lambda_j & \text{eigenvalue of } \mathbf{u}_j \\ \sigma^2 & \text{otherwise} \end{cases},$$

where **R** is $(D \times K)$ orthonomal matrix. The weighting term $\boldsymbol{\rho}$ allows to explicit control over the scale of the different projection axis. $\boldsymbol{\rho}$ controls if we should place more or less importance on the role of the input to the projection axis, which is meant to reflect our posterior belief of re-scaling due to not all data points sharing all subspaces.

C.2 Learning W

In this section, we compare two different schemes of learning the projection matrix **W** while maintaining orthogonality. We generate synthetic data $\mathbf{Y} \in \mathbb{R}^{D \times N}$ which takes the form $\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E}$ with $\mathbf{X} \in \mathbb{R}^{K \times N}$ a latent feature matrix with standard Gaussian distribution; $\mathbf{W} \in \mathbb{R}^{D \times K}$ is a projection matrix with orthogonal columns; $\mathbf{E} \in \mathbb{R}^{D \times N}$ is a noise matrix with multivariate Gaussian columns each with mean zero, and without the loss of generality the covariance matrix $\sigma^2 \mathbf{I}_D$ with $\sigma = 0.1$. The core of the generative model remains the same across the different data sets we generate ($N = 1000$, $D = 35$), and only the number of latent features K and the indicator matrix **Z**, changes. We have considered five separate synthetic data sets and the distribution of **Z** for each setup is displayed in Figure 5.2. We evaluate how three different schemes are able to learn the projection matrix **W**; the first method is the same approach proposed by [106] will sample each column of **W** using the von Misses-Fisher (vMF) distribution (and then re-scale to maintain orthogonality), the second method will jointly sample all the columns of **W** using a matrix vMF distribution, and the third method will jointly optimises all columns of **W** on the Stiefel manifold using the Pymanopt toolbox [111]. For each method we solely focus on learning **W**, therefore other parameters are fixed

Table C.1: Performance of two different schemes learning the projection matrix **W** on five different datasets of dimension $D = 35$ and latent features $K = 10$ & $K = 20$. For each experiment **W** was estimated using 80% of the data and tested on the remaining 20%; average and 1SD of mean square prediction error over 20 different experiments is reported.

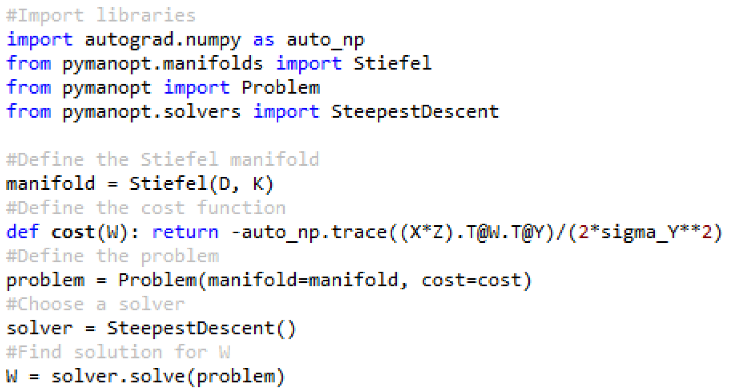
Data		Independent vMF [106]	Matrix vMF	Pymanopt
Sparse matrix	$K = 10$	0.323 ±0.016	0.221 ±0.012	**0.095** **±0.002**
	$K = 20$	0.471 ±0.013	0.324 ±0.009	**0.096** **±0.002**
Dense matrix	$K = 10$	0.548 ±0.026	0.102 ±0.020	**0.095** **±0.001**
	$K = 20$	0.793 ±0.017	0.623 ±0.025	**0.096** **±0.001**
Multivariate hypergeometric matrix	$K = 10$	0.563 ±0.029	0.345 ±0.050	**0.095** **±0.001**
	$K = 20$	0.760 ±0.017	0.498 ±0.012	**0.096** **±0.001**
Subsapce clustering	$K = 10$	0.566 ±0.020	0.127 ±0.035	**0.093** **±0.002**
	$K = 20$	0.769 ±0.010	0.125 ±0.023	**0.096** **±0.001**
Single State	$K = 10$	0.614 ±0.011	0.452 ±0.010	**0.093** **±0.001**
	$K = 20$	0.920 ±0.015	0.863 ±0.008	**0.096** **±0.003**

to the true value. For each experiment **W** is estimated using 80% of the data and then tested on the remaining 20%; results are reported in Table C.1.

Results from Table C.1 show that the projection matrix **W** estimated using the Pymanopt toolbox has a lower mean squared prediction error when compared to estimating it with the other two methods.

C.3 Projection matrix update using Pymanopt

For both the isPPCA and aPPCA, the matrix **W** is updated numerically by minimising the negative-log of equation (5.10) over the Stiefel manifold with respect to the matrix **W**. Figure C.1 shows the implementation of this using the Pymanopt toolbox [111].



```
#Import libraries
import autograd.numpy as auto_np
from pymanopt.manifolds import Stiefel
from pymanopt import Problem
from pymanopt.solvers import SteepestDescent

#Define the Stiefel manifold
manifold = Stiefel(D, K)
#Define the cost function
def cost(W): return -auto_np.trace((X*Z).T@W.T@Y)/(2*sigma_Y**2)
#Define the problem
problem = Problem(manifold=manifold, cost=cost)
#Choose a solver
solver = SteepestDescent()
#Find solution for W
W = solver.solve(problem)
```

Figure C.1: Python code for aPPCA updates on the rotation matrix **W** using PYMANOPT toolbox.

Appendix D

Bayesian nonparametric extensions supplementary material

D.1 Beta-binomial

Derivation of equation (6.1)

The the joint probability over $\mathbf{z}_k = \left[z_{1k}, ..., z_{Nk}\right]^T$ is:

$$\mathrm{P}\left(\mathbf{z}_k | \pi_k, m\right) \mathrm{P}\left(\pi_k | \alpha\right) = \left(\frac{(m!)^N}{\prod_n \left(m - z_{nk}\right)! \left(z_{nk}\right)!}\right) \left(\frac{\alpha}{K}\right) \left(1 - \pi_k\right)^{Nm - n_k} \left(\pi_k\right)^{n_k + \frac{\alpha}{K} - 1},$$

which when marginalised with respect to π_k, is:

$$\begin{aligned} \mathrm{P}\left(\mathbf{z}_k | \alpha, m\right) &= \left(\frac{(m!)^N}{\prod_n \left(r - z_{nk}\right)! \left(z_{nk}\right)!}\right) \left(\frac{\alpha}{K}\right) \int_0^1 \left(1 - \pi_k\right)^{Nm - m_k} \left(\pi_k\right)^{n_k + \frac{\alpha}{K} - 1} d\pi_k \\ &= \left(\prod_{n=1}^{N} \frac{(m!)}{\left(m - z_{nk}\right)! \left(z_{nk}\right)!}\right) \left(\frac{\alpha}{K}\right) \frac{\Gamma\left(n_k + \frac{\alpha}{K}\right) \Gamma\left(Nm - n_k + 1\right)}{\Gamma\left(\frac{\alpha}{K} + Nm + 1\right)}, \end{aligned}$$

Then the probability distribution over the matrix $\mathbf{Z} = \left[\mathbf{z}_1, ..., \mathbf{z}_K\right]$ is:

$$\mathrm{P}\left(\mathbf{Z} | \alpha, m\right) = \prod_{k=1}^{K} \left(\prod_{n=1}^{N} \frac{(m!)}{\left(m - z_{nk}\right)! \left(z_{nk}\right)!}\right) \left(\frac{\alpha}{K}\right) \frac{\Gamma\left(n_k + \frac{\alpha}{K}\right) \Gamma\left(Nm - n_k + 1\right)}{\Gamma\left(\frac{\alpha}{K} + Nm + 1\right)},$$

where $n_k = \sum_{n=1}^{N} z_{nk}$

Derivation of equation (6.2)

We assume there are K^+ number of columns of **Z** where $m_k > 0$, so we can split the above up:

$$\begin{aligned}
\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m\right) = &\frac{K!}{\prod_{h=0}^{(m+1)^N-1} D_h!}\left[\left(\frac{\alpha}{K}\right)\frac{\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nm+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}\right]^{K-K^+}\\
&\times\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\frac{m!}{(m-z_{nk})!\,(z_{nk})!}\right)\left(\frac{\alpha}{K}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nm-n_k+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}\\
=&\frac{K!}{\prod_{h=0}^{(m+1)^N-1} K_h!}\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nm+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}\right]^{K}\left[\frac{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nm+1\right)}\right]^{K^+}\\
&\times\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\frac{m!}{(m-z_{nk})!\,(z_{nk})!}\right)\left(\frac{\alpha}{K}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nm-n_k+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}\\
=&\frac{K!}{\prod_{h=0}^{(m+1)^N-1} K_h!}\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nm+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nm+1\right)}\right]^{K}\\
&\times\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\binom{m}{z_{nk}}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nm-n_k+1\right)}{\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nm+1\right)}.
\end{aligned}$$

Let us observe the following:

$$\frac{x\Gamma\left(x\right)}{\Gamma\left(N+1+x\right)} = \frac{1}{\prod_{n=1}^{N}\left(j+x\right)},$$

then:

$$\frac{\frac{\alpha}{K}\Gamma\left(n_k+\frac{\alpha}{K}\right)}{\frac{\alpha}{K}\Gamma\left(\frac{\alpha}{K}\right)} = \frac{\alpha}{K}\prod_{n=1}^{n_k-1}\left(j+\frac{\alpha}{K}\right) = \frac{\alpha}{K}\left(n_k-1+\frac{\alpha}{K}\right)!,$$

Then from $\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m\right)$, we get:

$$\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\binom{m}{z_{nk}}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr-n_k+1\right)}{\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)},$$

which becomes:

$$\prod_{k=1}^{K^+}\left(\prod_{n=1}^{N}\binom{m}{z_{nk}}\right)\frac{\frac{\alpha}{K}\left(n_k-1+\frac{\alpha}{K}\right)\Gamma\left(Nr-n_k+1\right)}{\Gamma\left(Nr+1\right)},$$

and:

$$\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nr+1\right)}\right]=\left[\frac{\Gamma\left(Nr+1\right)}{\left(Nr+\frac{\alpha}{K}\right)!}\right],$$

then:

$$\mathrm{P}\left(\left[\mathbf{Z}\right]|\alpha,m\right)=\frac{K!}{K_0!\prod_{h=1}^{(m+1)^N-1}K_h}\left[\frac{\Gamma\left(Nm+1\right)}{\left(Nm+\frac{\alpha}{K}\right)!}\right]^K \\ \times\left(\frac{\alpha}{K}\right)^{K^+}\prod_{k=1}^{K^+}\frac{\prod_{n=1}^{N}\binom{m}{z_{nk}}\left(n_k-1+\frac{\alpha}{K}\right)\Gamma\left(Nm-n_k+1\right)}{\Gamma\left(Nm+1\right)}.$$

Then as $K\to\infty$:

$$\lim_{K\to\infty}\left(\prod_{j=1}^{m_k-1}\left(j+\frac{\alpha}{K}\right)\right)=(m_k-1)!,$$

$$\lim_{K\to\infty}\left(\frac{K!}{K_0!K^{K^+}}\right)=1,$$

$$\lim_{K\to\infty}\left(\left[\frac{Nm!}{\prod_{j=1}^{Nm}\left(j+\frac{\alpha}{K}\right)}\right]^K\right)=\exp\left(-\alpha H_{Nm}\right),$$

then:

$$\lim_{K\to\infty}\left(\frac{K!}{K_0!\prod_{h=1}^{(m+1)^N-1}K_h}\left[\frac{\Gamma\left(Nm+1\right)}{\left(Nm+\frac{\alpha}{K}\right)!}\right]^K\left(\frac{\alpha}{K}\right)^{K^+} \\ \times\prod_{k=1}^{K^+}\frac{\prod_{n=1}^{N}\binom{m}{z_{nk}}\left(n_k-1+\frac{\alpha}{K}\right)\Gamma\left(Nm-n_k+1\right)}{\Gamma\left(Nm+1\right)}\right),$$

becomes:

$$\frac{\alpha^{K^+}}{\prod_{h=1}^{(m+1)^N-1}K_h}\exp\left(-\alpha H_{Nm}\right)\prod_{k=1}^{K^+}\frac{\prod_{n=1}^{N}\binom{m}{z_{nk}}\left(n_k-1\right)!\left(Nm-n_k\right)!}{(Nm)!},$$

where $H_{Nm}=\sum_{j=1}^{Nm}\frac{1}{j}$; also known as the (Nm)th harmonic number.

D.2 Beta-negative-binomial

Derivation of equation (6.3)

The the joint probability over $\mathbf{z}_k = [z_{1k}, ..., z_{Nk}]^T$is: The joint probability over $\mathbf{z}_k = [z_{1k}, \ldots, z_{Nk}]^T$is:

$$\mathrm{P}\left(\mathbf{z}_k|\pi_k, r\right)\mathrm{P}\left(\pi_k|\alpha\right) = \left[\prod_{n=1}^{N} \binom{z_{nk}+r-1}{z_{nk}}\right]\left(\frac{\alpha}{K}\right)\left(1-\pi_k\right)^{Nr}\left(\pi_k\right)^{n_k+\frac{\alpha}{D}-1},$$

where $n_k = \sum_{n=1}^{N} z_{nk}$. Which when marginalised with respect to π_k is:

$$\begin{aligned}\mathrm{P}\left(\mathbf{z}_k|\alpha, r\right) &= \left[\prod_{n=1}^{N} \binom{z_{nk}+r-1}{z_{nk}}\right]\left(\frac{\alpha}{K}\right)\int_0^1 \left(1-\pi_k\right)^{Nr}\left(\pi_k\right)^{n_k+\frac{\alpha}{D}-1} d\pi_k \\ &= \left[\prod_{n=1}^{N} \binom{z_{nk}+r-1}{z_{nk}}\right]\left(\frac{\alpha}{K}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)},\end{aligned}$$

Then the probability distribution over the matrix $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_K]$ is:

$$\mathrm{P}\left(\mathbf{Z}|\alpha, r\right) = \prod_{d=1}^{D}\left[\prod_{n=1}^{N} \binom{z_{nk}+r-1}{z_{nk}}\right]\left(\frac{\alpha}{K}\right)\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)}$$

where $n_k = \sum_{n=1}^{N} z_{nk}$

Derivation of equation (6.4)

We assume there are K^{+} number of columns of $\mathbf{Z}$ where $n_k > 0$, so we can split the above up:

$$
\begin{aligned}
\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m, r\right) = {} & \frac{K!}{\prod_{h=0}^{(m+1)^{N}-1} K_h!}\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)}\right]^{K-K^{+}} \\
& \times \prod_{k=1}^{K^{+}}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)} \\
= {} & \frac{K!}{\prod_{h=0}^{(m+1)^{N}-1} K_h!}\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nr+1\right)}\right]^{K}\left[\frac{\Gamma\left(\frac{\alpha}{K}+Nr+1\right)}{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}\right]^{K^{+}} \\
& \times \prod_{k=1}^{K^{+}}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)} \\
= {} & \frac{K!}{\prod_{h=0}^{(m+1)^{N}-1} K_h!}\left[\frac{\left(\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(Nr+1\right)}{\Gamma\left(\frac{\alpha}{K}+Nr+1\right)}\right]^{K} \\
& \times \prod_{k=1}^{K^{+}}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\Gamma\left(n_k+\frac{\alpha}{K}\right)\Gamma\left(\frac{\alpha}{K}+Nr+1\right)}{\Gamma\left(\frac{\alpha}{K}\right)\Gamma\left(n_k+\frac{\alpha}{K}+Nr+1\right)} \\
= {} & \frac{K!}{\prod_{h=0}^{(m+1)^{N}-1} K_h!}\left[\frac{\Gamma\left(Nr+1\right)}{\left(Nr+\frac{\alpha}{K}\right)!}\right]^{K}\left(\frac{\alpha}{K}\right)^{K^{+}} \\
& \times \prod_{k=1}^{K^{+}}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(n_k-1+\frac{\alpha}{K}\right)!\left(\frac{\alpha}{K}+Nr\right)!}{\left(n_k+\frac{\alpha}{K}+Nr\right)!},
\end{aligned}
$$

then:

$$
\begin{aligned}
\mathrm{P}\left([\mathbf{Z}]\,|\alpha, m, r\right) = {} & \frac{K!}{\prod_{h=0}^{(m+1)^{N}-1} K_h!}\left[\frac{\Gamma\left(Nr+1\right)}{\left(Nr+\frac{\alpha}{K}\right)!}\right]^{K}\left(\frac{\alpha}{K}\right)^{K^{+}} \\
& \times \prod_{k=1}^{K^{+}}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(n_k-1+\frac{\alpha}{K}\right)!\left(\frac{\alpha}{K}+Nr\right)!}{\left(n_k+\frac{\alpha}{K}+Nr\right)!},
\end{aligned}
$$

Then observe the following:

$$
\lim_{K\to\infty}\left(\prod_{j=1}^{n_k-1}\left(j+\frac{\alpha}{K}\right)\right) = (n_k-1)!,
$$

$$
\lim_{K\to\infty}\left(\frac{K!}{K_0!K^{K^{+}}}\right) = 1,
$$

$$\lim_{K\to\infty}\left(\left[\frac{Nr!}{\prod_{j=1}^{Nr}\left(j+\frac{\alpha}{K}\right)}\right]^{K}\right)=\exp\left(-\alpha H_{Nr}\right),$$

then:

$$\lim_{K\to\infty}\left(\frac{K!}{\prod_{h=0}^{(m+1)^N-1}K_h!}\left[\frac{\Gamma\left(Nr+1\right)}{\left(Nr+\frac{\alpha}{K}\right)!}\right]^{K}\left(\frac{\alpha}{K}\right)^{K^+}\right.$$
$$\left.\prod_{k=1}^{K^+}\left[\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right]\frac{\left(n_k-1+\frac{\alpha}{K}\right)!\left(\frac{\alpha}{K}+Nr\right)!}{\left(n_k+\frac{\alpha}{K}+Nr\right)!}\right),$$

is:

$$\frac{\alpha^{K^+}}{\prod_{h=1}^{(m+1)^N-1}K_h}\exp\left(-\alpha H_{Nr}\right)\prod_{k=1}^{K^+}\frac{\left(\prod_{n=1}^{N}\binom{z_{nk}+r-1}{z_{nk}}\right)\left(n_k-1\right)!\left(Nr\right)!}{\left(n_k+Nr\right)!},$$

where $H_{Nr}=\sum_{j=1}^{Nr}\frac{1}{j}$; also known as the (Nr)th harmonic number.