Nonparametric Bayesian Discrete Latent Variable Models for Unsupervised Learning

vorgelegt von
Dipl.-Ing. Dilan Görür
aus Seydişehir

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

Promotionsausschuß:
Vorsitzender: Prof. Dr. H. U. Heiß
Berichter: Prof. Dr. K. R. Müller
Berichter: Prof. Dr. M. Opper

Tag der wissenschaftlichen Aussprache: 20.04.2007

Berlin 2007
D83

Zusammenfassung

Die Analyse praktischer Fragestellungen erfordert oft Modelle, die robust und zugleich flexibel genug sind um Abhängigkeiten in den Daten präzise darzustellen. Nichtparametrische Bayesianische Modelle erlauben die Konstruktion solcher Modelle und können daher für komplexe Aufgaben herangezogen werden.

Unter nichtparametrischen Modellen sind dabei solche mit undendlich vielen Parametern zu verstehen. Die vorliegende Doktorarbeit untersucht zwei Varianten solcher Modelle: zum einen „Latent Class Models“ mit unendlich vielen latenten Klassen, und andererseits „Discrete Latent Feature Models“ mit unendlich vielen latenten Merkmalen. Für erstere verwenden wir Dirichlet Prozess-Mixturen (Dirichlet Process Mixtures, DPM) und für letztere den Indian Buffet-Prozess (IBP), eine Verallgemeinerung der DPM-Modelle.

Eine analytische Behandlung der in dieser Arbeit diskutierten Modelle ist nicht möglich, was approximative Verfahren erforderlich macht. Bei solchen Verfahren kann die Verwendung geeigneter konjugierter a priori Verteilungen zu bedeutenden Vereinfachungen führen. Im Rahmen komplexer Modelle stellen solche Verteilungen allerdings oft eine zu starke Beschränkung dar. Ein Hauptthema dieser Arbeit sind daher Markov-Ketten Monte Carlo (MCMC) Verfahren zur approximativen Inferenz, die auch ohne konjugierte a priori Verteilung effizient einsetzbar sind.

In Kapitel 2 definieren wir grundlegende Begriffe und erklären die in dieser Arbeit verwendete Notation. Der Dirichlet-Prozess (DP) wird in Kapitel 3 eingeführt, zusammen mit einigen unendlichen Mixturmodellen, welche diesen als a priori Verteilung verwenden. Zunächst geben wir einen Überblick über bisherige Arbeiten zur Definition eines Dirichlet-Prozesses und beschreiben die MCMC Techniken, die zur Behandlung von DPM-Modellen entwickelt wurden. DP Mixturen von Gaußverteilungen (Dirichlet process mixtures of Gaussians, DPMoG) wurden vielfach zur Dichteschätzung eingesetzt. Wir zeigen eine empirische Studie über die Abwägung zwischen analytischer Einfachheit und Modellierungsfähigkeit bei der Verwendung konjugierter a priori Verteilungen im DPMoG. Die Verwendung von bedingt konjugierten im Gegensatz zu konjugierten a priori Verteilungen macht weniger einschränkende Annahmen, was ohne eine deutliche Erhöhung der Rechenzeit zu besseren Schätzergebnissen führt. In einem Faktor-Analyse-Modell wird eine Gaußverteilung durch eine spärlich parametrisierte Kovarianzmatrix repräsentiert. Wir betrachten eine Mixtur solcher Modelle (mixture of factor analyzers, MFA), wobei wiederum die Anzahl der Klassen nicht beschränkt ist (Dirichlet Process MFA, DPMFA). Wir benutzen DPMFA, um Aktionspotentiale verschiedener Neuronen aus extrazellulären Ableitungen zu gruppieren (spike sorting).

Kapitel 4 behandelt Indian Buffet Prozesse (IBP) und unendliche latente Merkmalsmodelle mit IBPs als a priori Verteilungen. Der IBP ist eine Verteilung über binäre

Matrizen mit unendlich vielen Spalten. Wir beschreiben verschiedene Ansätze zur Konstruktion von IBPs und stellen einige neue MCMC Verfahren zur approximativen Inferenz in Modellen dar, die den IBP als a priori Verteilung benutzen. Im Gegensatz zur etablierten Methode des „Gibbs Sampling“ haben unsere Verfahren den Vorteil, dass sie keine konjugierten a priori Verteilungen voraussetzen. Bei einem vorgestellten empirischen Vergleich liefern sie dennoch ebenso gute Ergebnisse wie Gibbs Sampling. Wir zeigen außerdem, dass ein nichtkonjugiertes IBP Modell dazu in der Lage ist, die latenten Variablen handgeschriebener Ziffern zu lernen. Ferner benutzen wir eine IBP a priori Verteilung, um eine nichtparametrische Variante des „Elimination-by-aspects“ (EBA) Auswahlmodells zu formulieren. Eine vorgestellte Paar-Vergleichs-Studie demonstriert dessen präzise Vorhersagen des menschlichen Auswahlverhaltens.

Abstract

The analysis of real-world problems often requires robust and flexible models that can accurately represent the structure in the data. Nonparametric Bayesian priors allow the construction of such models which can be used for complex real-world data.

Nonparametric models, despite their name, can be defined as models that have infinitely many parameters. This thesis is about two types of nonparametric models. The first type is the latent class models (i.e. a mixture model) with infinitely many classes, which we construct using Dirichlet process mixtures (DPM). The second is the discrete latent feature models with infinitely many features, for which we use the Indian buffet process (IBP), a generalization of the DPM.

Analytical inference is not possible in the models discussed in this thesis. The use of conjugate priors can often make inference somewhat more tractable, but for a given model the family of conjugate priors may not always be rich enough. Methodologically this thesis will rely on Markov chain Monte Carlo (MCMC) techniques for inference, especially those which can be used in the absence of conjugacy.

Chapter 2 introduces the basic terminology and notation used in the thesis. Chapter 3 presents the Dirichlet process (DP) and some infinite latent class models which use the DP as a prior. We first summarize different approaches for defining the DP, and describe several established MCMC algorithms for inference on the DPM models. The Dirichlet process mixtures of Gaussians (DPMoG) model has been extensively used for density estimation. We present an empirical comparison of conjugate and conditionally conjugate priors in the DPMoG, demonstrating that the latter can give better density estimates without significant additional computational cost. The mixtures of factor analyzers (MFA) model allows data to be modeled as a mixture of Gaussians with a reduced parametrization. We present the formulation of a nonparametric form of the MFA model, the Dirichlet process MFA (DPMFA). We utilize the DPMFA for clustering the action potentials of different neurons from extracellular recordings, a problem known as spike sorting.

Chapter 4 presents the IBP and some infinite latent feature models which use the IBP as a prior. The IBP is a distribution over binary matrices with infinitely many columns. We describe different approaches for defining the distribution and present new MCMC techniques that can be used for inference on models which use it as a prior. Empirical results on a conjugate model are presented showing that the new methods perform as well as the established method of Gibbs sampling, but without the requirement for conjugacy. We demonstrate the performance of a non-conjugate IBP model by successfully learning the latent features of handwritten digits. Finally, we formulate a nonparametric version of the elimination-by-aspects (EBA) choice model using the IBP, and show that it can make accurate predictions about the people's choice outcomes in a paired comparison task.

Contents

Zusammenfassung iii

Abstract v

List of Algorithms ix

Acknowledgments xi

Notation xiii

1 Introduction 1

2 Nonparametric Bayesian Analysis 3

3 Dirichlet Process Mixture Models 9

3.1 The Dirichlet Process 11
3.1.1 Pólya's Urn 12
3.1.2 Chinese Restaurant Process 14
3.1.3 DP as a Normalized Gamma Process 16
3.1.4 Stick Breaking Construction 17
3.1.5 Infinite Mixture Models 19
3.1.6 Properties of the Distribution 21
3.2 MCMC Inference in Dirichlet Process Mixture Models 24
3.2.1 Algorithms for Conjugate Models using the Pólya Urn Scheme . 25
3.2.2 Algorithms for non-Conjugate DP Models 29
3.2.3 Algorithms Using the Stick-Breaking Representation 34
3.3 Empirical Study on the Choice of the Base Distribution 40
3.3.1 The Dirichlet Process Gaussian Mixture Model 40
3.3.2 Inference Using Gibbs Sampling 43
3.3.3 Experiments 47
3.3.4 Conclusions 52
3.4 Dirichlet Process Mixtures of Factor Analyzers 52
3.4.1 Spike Sorting Using DPMFA 56
3.4.2 Experiments 57
3.4.3 Conclusions 62
3.5 Discussion 65

4 Indian Buffet Process Models **67**
4.1 The Indian Buffet Process . . . 68
4.1.1 A Distribution on Infinite Binary Matrices . . . 71
4.1.2 Stick-Breaking Construction . . . 74
4.1.3 A Special Case of the Beta Process . . . 78
4.1.4 Properties of the Distribution on the Infinite Binary Martices . . 79
4.2 MCMC Sampling algorithms for IBLF models . . . 81
4.2.1 Gibbs Sampling for Conjugate IBLF models . . . 83
4.2.2 Approximate Gibbs Sampling for Non-Conjugate Models . . . 85
4.2.3 Metropolis-Hastings Sampling . . . 86
4.2.4 Gibbs Sampling for the Truncated Stick-Breaking Construction . 88
4.2.5 Slice Sampling Using the Stick Breaking Construction . . . 90
4.2.6 Slice Sampling for the Semi-Ordered Stick-Breaking . . . 93
4.3 Comparing Performances of the Samplers . . . 95
4.4 A Flexible Infinite Latent Feature Model . . . 98
4.5 A Choice Model with Infinitely Many Latent Features . . . 100
4.5.1 Model Specification . . . 103
4.5.2 Inference using MCMC . . . 105
4.5.3 Experiments . . . 105
4.5.4 Conclusions . . . 111
4.6 Discussion . . . 111

5 Conclusions **113**

A Details of Derivations for the Stick-Breaking Representation of IBP **115**
A.1 Densities of ordered stick lengths . . . 115
A.2 Probability of a part of Z being inactive . . . 116

B Mathematical Appendix **119**
B.1 Dirichlet Distribution . . . 119
B.2 Parameterization of the Distributions Used . . . 122
B.3 Definitions . . . 122
B.4 Equalities and Limits . . . 123

Bibliography **125**

List of Algorithms

1 Gibbs sampling for conjugate DPM models with full parameter representation. 26

2 Gibbs sampling for conjugate DPM models using indicator variables and component parameters. 27

3 Gibbs sampling for conjugate DPM models using indicator variables without parameter representations. 28

4 The No-Gaps algorithm for non-conjugate DPM models. 29

5 Gibbs sampling for non-conjugate DPM models using auxiliary components 31

6 Metropolis-Hastings sampling with restricted Gibbs updates for non-conjugate DPM models 32

7 Auxiliary variable sampling for non-conjugate DPM models. 33

8 Gibbs sampling for truncated DP 35

9 Retrospective sampling for DPM models using stick-breaking construction 37

10 Slice sampling algorithm for the DPM model 39

11 Gibbs sampling for conjugate IBP 85

12 Approximate Gibbs sampling for non-conjugate IBP 87

13 Metropolis-Hastings sampling for IBP 89

14 Gibbs sampling for truncated IBP 90

15 Slice sampling for stick-breaking IBP 93

16 Slice sampling for the semi-ordered IBP 96

Acknowledgements

I was fortunate enough to do my PhD in a great research atmosphere in the Department of Empirical Inference for Machine Learning and Perception at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany. I would like to thank Berhard Schölkopf for providing this excellent atmosphere and giving me the opportunity to be a part of it. I am indebted to my adviser Carl Edward Rasmussen for introducing me to the Bayesian framework and for sharing his enthusiasm. I wish to thank for his guidance, support and motivation during my research.

Moreover, I would like to thank my committee members for the useful discussions and comments. I am especially thankful to my ”Doktorvater” Klaus Robert Müller for his help during my PhD and his group for the friendly welcome in Berlin.

It was a pleasure to work and live in Tübingen. The people I met and worked with during my time here including past and present members and visitors of AGBS are what made my work possible. Their scientific input, discussions, support, and friendship were invaluable. It is not possible to acknowledge them all here but I would like to thank especially to Malte Kuss, HyunJung Shin, Jeremy Hill, Fabian Sinz, Arthur Gretton, Philipp Berens, Frank Jäkel, Matthias Seeger, Olivier Chapelle, Joaquin Quiñonero Candela, Lehel Csato, Jan Eichorn, Navin Lal, Tobias Pfingsten, Yee Whye Teh and Andreas Tolias for their scientific inputs and their friendship. Furthermore, I would like to thank Sabrina Nielebock and Sebastian Stark for their administrative support.

I would like to thank Frank Jäkel, Arthur Gretton, Matthias Seeger and Jan Eichorn for reading parts of the manuscript and providing comments on the earlier versions and Steffi Jegelka for helping with the ”Zusammenfassung”.

Finally, a special thanks to my friends Elif, Sven and Doug for cheering me up.

Notation

Matrices are capitalized and vectors are in bold type. We do not generally distinguish between probabilities and probability densities.

Abbreviation	**Meaning**
cdf	Cumulative distribution function
i.i.d.	Independently and identically distributed
pdf	Probability density function
BDMCMC	Birth-and-Death MCMC
BTL	Bradley-Terry-Luce
CCDP	DPMoG model with conditionally conjugate base distribution
CDP	DPMoG model with fully conjugate base distribution
CRP	Chinese restaurant process
DP	Dirichlet Process
DPM	Dirichlet Process mixture
DPMFA	Dirichlet Process Mixtures of Factor Analyzers
DPMoG	Dirichlet Process Mixture of Gaussians
EBA	Elimination by Aspects
FA	Factor Analysis
IBLF	Infinite binary latent features
IBP	Indian buffet process
MCMC	Markov chain Monte Carlo
MFA	Mixtures of Factor Analyzers
MoG	Mixture of Gaussians
PCA	Principle component analysis
RJMCMC	Reversible Jump MCMC

Symbol	**Meaning**
general	
$\propto$	proportional to; e.g. $p(x) \propto f(x)$ means $p(x)$ is equal to $f(x)$ times a factor that is independent of x
$\sim$	distributed according to; e.g. $\mathbf{x} \sim F(\mathbf{x} \mid \theta)$ means $\mathbf{x}$ has distribution $F(\theta)$
$\otimes$	elementwise multiplication
$\delta_\theta(\cdot)$	probability measure concentrated at θ
$A \backslash B$	set difference
D	dimension of input space
$D_{\mathrm{KL}}(p \Vert q)$	the KL divergence between the density p and q
$\mathrm{E}\big\{f(\phi)\big\}$	expectation of function f taken with respect to the distribution of ϕ
H_N	Nth harmonic number, $H_N = \sum_{i=1}^N 1/i$
$\mathbb{I}(A)$	the indicator function for a measurable set A; $\mathbb{I}(A) = 1$ if A is true, 0 otherwise
$L(\hat{\phi}, \phi)$	loss encountered by predicting $\hat{\phi}$ when the true value is ϕ
$\mathcal{L}(X \mid \Theta)$	likelihood of the parameter(s) Θ for the data points X
N	number of training points
$\mathbf{x}_i$	ith data point
X	data matrix containing the set of observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
DP	
$\mathcal{D}(\alpha_1, \ldots, \alpha_k)$	k-dimensional Dirichlet distribution
$DP(\alpha, G_0)$	Dirichlet process with concentration parameter α and base distribution G_0
c_i	indicator variable for $\mathbf{x}_i$ showing the component assignment
$\mathbf{c}$	set of all indicator variables $\{c_1, \ldots, c_N\}$
$\{\theta_i\}_1^n$	sequence of random variables $\theta_i, \quad i = 1, \ldots, n$
θ_i	parameter associated with $\mathbf{x}_i$
θ_{-i}	set of all parameters other than i
ϕ_k	parameter associated with kth component (feature)
π_k	mixing proportion for the kth mixture component
n_k	the number of data points assigned to component k
$n_{\cdot<i,k}$	the number of data points assigned to component k before the ith data point
K	the number of components in a finite mixture model,
$K^\ddagger$	the number of active components
$K^\dagger$	the number of represented components
v_k	variable that determines the breaking point of the stick at kth iteration for DP

Symbol	**Meaning**
IBP	
Z	binary latent feature matrix
z_{ik}	an entry of matrix Z
$\mathbf{z}_i$	ith row of Z, features of ith customer
$\mathbf{z}_k$	kth column of Z, kth feature vector
$[Z]$	equivalence class of binary matrix Z
m_k	the number of customers that has the kth dish
$m_{.<i,k}$	the number of customers that sampled the kth dish before the ith customer
K	the number of components in a finite mixture model,
$K^{(i)}$	the number of existing unique dishes for the ith customer
$K_*^{(i)}$	the number of new unique dishes sampled by the ith customer
$K^\ddagger$	the number of active features
$K^\dagger$	the number of represented features
$K_{kj}^\dagger$	the index of the last active component after changing the component assignment of $\mathbf{x}_i$ from k to j
$\mathit{lof}(\cdot)$	*left-ordered form* function to map binary matrices to their equivalence classes
h	history, the binary number corresponding to a binary column vector in Z
K_h	the number of columns with history h
μ_k	Feature presence probability for column k
$\mu_{(k)}$	kth largest value among $\mu_1, \ldots, \mu_K$, feature presence probability for the kth feature in the stick-breaking construction
$\mu_{(1:k)}$	First k largest values of $\mu_1, \ldots, \mu_K$
μ_k^+	feature presence probability of the kth active feature
μ_k°	feature presence probability of the kth represented empty feature
$\mathbf{L}_k$	the set of indices other than the indices of k largest μ's, $\{1, \ldots, K\} \backslash \{l_1, \ldots, l_k\}$
ν_k	variable that determines the breaking point of the stick at kth iteration for IBP
p_{ij}	probability of choosing alternative i over alternative j
x_{ij}	number of times the alternative i was chosen over alternative j in a choice scenario

1 Introduction

A statistical model aims to represent the observed data with the goal of obtaining comprehensible summaries of it and making predictions about the future observations. Bayesian models seek to model the process that generated the observations, taking into account the prior belief about the generative process. Bayesian models have the advantage of making the model assumptions explicit and providing predictions with uncertainties. The prior belief about the model parameters is expressed by the prior distributions on the parameters, possibly in a hierarchical manner.

The analysis of complex real world data requires flexible models that are robust to outliers and that can represent distributions which are not of standard form. Nonparametric models allow definition of flexible models which can successfully describe the variability in the data. Nonparametric models are traditionally described to be models that have infinite dimensional parameters. The focus of this thesis will be on two types of nonparametric models: mixture models with infinitely many components and binary latent feature models with infinitely many features. Using conjugate priors typically makes inference easier however restricts the flexibility of the models. The emphasis of this thesis is on nonparametric models with non-conjugate priors and inference techniques for these complex models.

The Dirichlet process (DP) (Ferguson, 1973) is one of the most widely used nonparametric Bayesian distributions. The DP is a distribution over distributions. Thus, it can be used as a prior over the distribution of random variables. This provides a means of expressing the uncertainty about the distributional form of the parameters in a model. The models in which the DP is used as a prior over the distribution of the parameters are referred to as the Dirichlet process mixture (DPM) models. We will use the DP for defining mixture models with infinitely many components for density estimation and clustering.

Indian buffet process (IBP) (Griffiths and Ghahramani, 2005) is a distribution over infinite binary sparse matrices. The IBP is a generalization of the Chinese restaurant process (CRP) which is related to the DP. We will use IBP as a prior over the latent features for defining nonparametric latent feature models.

Chapter 3 starts with describing several different approaches for defining the Dirichlet process and explaining various inference techniques for the DPM models using Markov chain Monte Carlo (MCMC) techniques. After the overview of the definition and properties of the DP and the inference techniques, we consider the Dirichlet process mixtures of Gaussians model (DPMoG) for density estimation and clustering.

The DP is defined by two parameters, the base distribution and the concentration parameter. The base distribution can be interpreted as the prior guess for the random distribution function and the concentration parameter as expressing the strength of the belief in the prior. The specification of the base distribution for the DP, which corresponds to the prior on the component parameters for infinite mixture models, is often guided by mathematical and practical convenience. For DPM models, the use of conjugate priors makes the analysis much more tractable, however conjugate priors might fail to represent the prior beliefs. Specifically for the DPMoG model, the conjugate priors have some unappealing properties with prior dependencies between the mean and the covariance. Empirical assessment of the trade off between modeling performance and computational cost encountered when using conjugate priors for DPMoG is one of the primary goals of this thesis. In Section 3.3 we compare the DPMoG model with a conjugate and a conditionally-conjugate base distribution in terms of modeling performance and computational feasibility. It is possible to integrate out some of the parameters in the conditionally conjugate model, which vastly improves mixing. We show that this improvement makes possible the practical use of the more flexible conditionally conjugate model.

Mixtures of factor analyzers (MFA) is a mixture model that has Gaussian components with constrained covariance matrices. Assuming that most of the information lies in a lower dimensional space, high dimensional data can be efficiently modeled using this reduced parametrization. In Section 3.4 we define the Dirichlet process mixtures of factor analyzers (DPMFA) model and apply it to a challenging problem of clustering neuronal data, known as spike sorting.

The second group of nonparametric models we consider is the latent feature models with infinite number of binary features. The IBP is a distribution over infinite binary sparse matrices that has many correspondences to the DP. Using IBP as a prior over the latent features, we can define nonparametric latent feature models. Chapter 4 starts with the different approaches for defining the distribution induced by the IBP, and discusses the parallels to the DP. Section 4.2 describes different MCMC algorithms for inference on IBP models. The sampling algorithms are compared in Section 4.3 to give an intuition about their general performance. We demonstrate the modeling capability of the IBP models for learning latent features of handwritten digits in Section 4.4.

The elimination by aspects (EBA) model is an interesting choice model in which the latent variables are used to represent the features of the alternatives in the choice set that lead to the choice probabilities. In Section 4.5, we define the EBA model with IBP prior on the latent features and infer the choice probabilities by learning the latent features using this model. We conclude the thesis with a discussion in Chapter 5.

The next chapter gives a brief overview of Bayesian modeling to introduce the terminology and the notation that will be used in this thesis. Some mathematical formulas and statistical definitions are given in Appendix B.

2 Nonparametric Bayesian Analysis

A statistical model tries to explain the data in terms of the properties of the system that generated it. Probabilistic models assume that the data have been generated from an unknown probability distribution which may or may not follow a general parametric form. The model structure $\mathcal{M}$ is defined in terms of random variables, referred collectively as the parameters, Ψ. The model structure and the parameters are chosen such that the model can accurately represent the generative process that gave rise to the observed data. The Bayesian approach treats the parameters as being random quantities and therefore involves placing distributions over them, representing the prior belief. The prior is updated in light of the observations, giving the posterior. Below, we give a broad overview of Bayesian analysis. For details, refer to for example (Bernardo and Smith, 1994; Gelman et al., 2003; O'Hagan, 1994; Box and Tiao, 2003).

We denote the *prior* distribution of the parameters by $P(\Psi \,|\, \mathcal{M})$. The distribution of the data assumed by the model $P(\mathcal{D} \,|\, \Psi, \mathcal{M})$ is referred to as the *sampling distribution*. The *likelihood function* is the probability density of the *observed* data X conditioned on the unknown model parameters Ψ, therefore it is a function of Ψ, $\mathcal{L}(\Psi \,|\, \mathcal{M}) = P(X \,|\, \Psi, \mathcal{M})$. The Bayes' rule yields the *posterior* density:

$$P(\Psi \,|\, X, \mathcal{M}) = \frac{P(\Psi \,|\, \mathcal{M}) P(X \,|\, \Psi, \mathcal{M})}{P(X \,|\, \mathcal{M})}. \tag{2.1}$$

The denominator $P(X \,|\, \mathcal{M})$, obtained by integrating over the parameters,

$$P(X \,|\, \mathcal{M}) = \int P(\Psi \,|\, \mathcal{M}) P(X \,|\, \Psi, \mathcal{M}) \, \mathrm{d}\Psi, \tag{2.2}$$

is referred to as the *evidence* or the *marginal likelihood*. Note that this quantity does not depend on the parameters and it only appears as a normalizing constant for the posterior distribution of Ψ. Therefore, we generally write[1]

$$P(\Psi \,|\, X) \propto P(\Psi) P(X \,|\, \Psi), \tag{2.3}$$

meaning the posterior for the parameters is proportional to the prior times the likelihood. Thus, the posterior distribution expresses the updated belief about the parameters Ψ after observing data.

A family of prior distributions $\mathcal{F}$ is said to be *conjugate* to the likelihood if the posterior is also in $\mathcal{F}$. Using conjugate priors, the integral in eq. (2.2) can be analytically evaluated. However, the conjugate family is not rich enough to always match the prior

[1] In the following, we drop the conditioning on the model structure $\mathcal{M}$ from the notation.

belief. In this case, one would need to use priors from a larger family. Using a prior distribution from a more general family will typically result in the integral in eq. (2.2) being intractable, hence increased computational complexity in posterior calculations. *Conditionally conjugate* priors provide a richer family of distributions while retaining some of the tractability.

Bayesian inference refers to obtaining the posterior distributions for the parameters of interest and extracting information about these parameters from the posterior. After updating our beliefs about the model parameters using the Bayes' rule, we can use the model to make predictions about new observations $\mathbf{x}^*$, or about any unobserved quantity ϕ whose distribution depends on the parameters, using the *predictive distribution*,

$$P(\phi \,|\, X) = \int P(\phi \,|\, \Psi) P(\Psi \,|\, X) \, \mathrm{d}\Psi. \tag{2.4}$$

We may be asked to give a single prediction value $\hat{\phi}$, referred to as a *point estimate*, rather than the distribution of the predictions. In this case, we choose a value $\hat{\phi}$ from the predictive distribution that minimizes the expected loss for a given loss function $L(\hat{\phi}, \phi)$,

$$\mathrm{E}\big\{L(\hat{\phi}, \phi)\big\} = \int L(\hat{\phi}, \phi) \, P(\phi \,|\, X) \, \mathrm{d}\phi. \tag{2.5}$$

The optimal choice $\hat{\phi}$ is referred to as the *Bayes estimate.*

In density modeling, we want to model the generating density in the light of the observations. The Kullback-Leibler (KL) divergence is a standard measure of the discrepancy between two distributions. The difference of the estimated density $Q(\phi)$ to the true generating density $P(\phi)$ is given as

$$D_{\mathrm{KL}}(P \| Q) = \int P(\phi) \, \log \frac{P(\phi)}{Q(\phi)} \, \mathrm{d}\phi. \tag{2.6}$$

Since the generating density P is fixed, maximizing the log-likelihood minimizes the KL divergence.

Some other widely used loss functions are the squared error loss, $L(\hat{\phi}, \phi) = \big(\phi - \hat{\phi}\big)^2$ which is minimized by the posterior mean, the absolute error loss $L(\hat{\phi}, \phi) = \big|\phi - \hat{\phi}\big|$ minimized by the posterior median, and the zero-one loss,

$$\begin{aligned} L(\hat{\phi}, \phi) &= 1 \qquad \text{if} \big|\phi - \hat{\phi}\big| > \varepsilon \\ L(\hat{\phi}, \phi) &= 0 \qquad \text{otherwise,} \end{aligned} \tag{2.7}$$

minimized by the posterior mode.

Assumptions on the Model Structure

Above, we referred to the set of all unknown variables in a model as the parameters, denoted by Ψ. Some of the parameters in a model may have physical interpretations and therefore their values may be of interest, whereas some are merely necessary for building

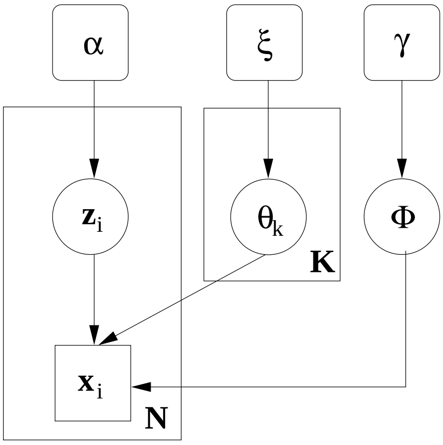

Figure 2.1: Graphical representation of a general generative model with latent variables $\mathbf{z}_i$ and parameters $\theta_1, \ldots, \theta_K$ and Φ. The observed quantities are shown in squares. The unknown variables are shown in circles, their posterior distributions are to be inferred. The parameters of these variables are given in the higher level. Their values may be fixed by prior knowledge, or the hierarchy can be extended by specifying hyperpriors. Large rectangles with numerals in the lower left hand corner denote replicated structures.

the model structure and are referred to as the *nuisance* parameters. Additionally, *auxiliary* parameters can be used which do not play any role in describing the model structure but assist the inference. All unknown variables in a model can said to be latent (or hidden) since they are not observed. But the term *latent variable* is generally used to refer to the variables number of which depends on the number of data points.

Generally, the order in which the data points are observed is not important. Therefore, the data points are assumed to be independent conditional on the underlying distribution, referred equivalently as the data being exchangeable. More precisely, the joint distribution of independently and identically distributed (i.i.d.) random variables can be written as a product of their distribution functions:

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \Phi) = \prod_{i=1}^{N} P(\mathbf{x}_i \mid \Phi). \tag{2.8}$$

We use the graphical representation depicted in Figure 2.1 to show the general hierarchical structure of a generative model. The set of observations $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are modeled with with latent variables $\mathbf{z}_i,\ i = 1, \ldots, N$, parameters Φ and $\theta_k,\ k = 1, \ldots, K$. We denote the replicated structures that are i.i.d. conditionally on some variables using the large rectangles with numerals in the lower left hand corner.

The term *parametric model* refers to the models in which the model has a form that is expressed by a finite number of parameters. The likelihood $P(X \mid \Psi)$ may be assumed to be of a known simple form such that obtaining the posterior distributions of interest is straightforward. It is possible to make the model more flexible by using a hierarchy and defining hyper-priors on the priors of the parameters. However, the assumed form of the distribution may be too restrictive such that the model fails to accurately represent the data.

Mixture models allow modeling distributions that do not belong to a parametric family, by representing the distribution in terms of $K > 1$ distributions of known simple form. Mixture modeling is a strong tool of probabilistic models since arbitrary distributions can be successfully modeled using simple distributions. Deciding on the number of components in the model is generally considered as a model selection problem. It is possible to treat the number of components, K, also as a parameter of the model and infer it from the observations (Richardson and Green, 1997; Stephens, 2000).

An alternative to parametric models is the *nonparametric models* which are models with (countably) infinitely many parameters. Nonparametric models achieve high flexibility and robustness by defining the prior to be a nonparametric distribution from a space of all possible distributions. All parameters in a model may be assumed to have a nonparametric prior distribution or the nonparametric prior can be defined over only a subset of the parameters, resulting in fully nonparametric and semiparametric models, respectively. In this thesis, we refer to all models with a nonparametric part as nonparametric models regardless of the presence (or absence) of other parameters with parametric prior distributions.

In spite of having infinitely many parameters, inference in the nonparametric models is possible since only a finite number of parameters need to be explicitly represented. This brings close the models with finite but unknown number of components and the nonparametric models. However, there are conceptual and practical differences between the two. For instance, to learn the dimensionality of the parametric model we need to move between different model dimensions (Green, 1995; Cappé et al., 2003). On the other hand, the nonparametric model always has infinitely many parameters although many of them do not need to be represented in the computations. As a consequence, the posterior of the parametric model is of known finite dimension whereas the posterior of a nonparametric model is also nonparametric, that is, it is also represented by infinitely many parameters. We will focus on nonparametric models in this work. For more details on Bayesian nonparametric methods, refer to for example (Ferguson et al., 1992; Dey et al., 1998; Walker et al., 1999; MacEachern and Müller, 2000; Gosh and Ramamoorthi, 2003; Müller and Quintana, 2004; Rasmussen and Williams, 2006).

Approximate Inference Techniques

For some simple models, it is possible to calculate the posterior distribution of interest analytically, however this is not the case generally. The specific form of prior for which the integral over the parameters can be evaluated to get the marginal likelihood is referred to as the conjugate prior for the likelihood. Often, the integrals we need to compute to get the posterior distribution is not tractable, therefore we need approximation techniques for inference.

Approximate inference methods include Laplace approximation, variational Bayes, mean field approximation, expectation maximization and Markov chain Monte Carlo (MCMC). We will use MCMC for inference in this thesis. For an introduction to the MCMC methods refer to for example Neal (1993) and Gilks et al. (1995).

3 Dirichlet Process Mixture Models

Bayesian inference requires assigning prior distributions to all unknown quantities in a model. In parametric models, the prior distributions are assumed to be of known parametric form, specified by hyperparameters. Hierarchical models are used when there is uncertainty about the *value* of the hyperparameters. The uncertainty about the *parametric form* of the prior distribution can be expressed by specifying a prior distribution on the distribution functions using Bayesian nonparametrics. There have been many Bayesian nonparametric priors developed, see for example (Freedman, 1963; Ferguson, 1974; Sibisi and Skilling, 1997). The Dirichlet process (DP) is one of the most prominent random probability measures due to its richness, computational ease, and interpretability.

The DP, first introduced by Ferguson (1973), can be defined using several different perspectives. Ferguson (1973) uses Kolmogorov consistency conditions to define the DP and also gives an alternative definition using the gamma process. Blackwell and MacQueen (1973) use exchangeability to show that a generalization of the Pólya urn scheme leads to the DP. A closely related sequential process is the Chinese restaurant process (CRP) (Aldous, 1985; Pitman, 2006), a distribution over partitions, which also results in the DP when each partition is assigned an independent parameter with a common distribution. A constructive definition of the DP was given by Sethuraman and Tiwari (1982), which leads to the characterization of the DP as a stick-breaking prior (Ishwaran and James, 2001). Additionally, the DP can be represented as a special case of many other random measures, see (Doksum, 1974; Pitman and Yor, 1997; Walker et al., 1999; Neal, 2001).

The DP is defined by two parameters, a positive scalar α and a probability measure G_0, referred to as the concentration parameter and the base measure, respectively.

The base distribution G_0 is the parameter on which the nonparametric distribution is centered, which can be thought of as the prior guess (Antoniak, 1974). The concentration parameter α expresses the strength of belief in G_0. For small values of α, samples from a DP is likely to be composed of a small number of atomic measures. For large values, samples are likely to be *concentrated* around G_0.

The hierarchical models in which the DP is used as a prior over the distribution of the parameters are referred to as the Dirichlet Process mixture (DPM) models, also called mixture of Dirichlet process models by some authors due to Antoniak (1974). Putting a prior G on the distribution of model parameters θ, gives the following model:

$$\begin{aligned} \mathbf{x}_i \,|\, \theta_i &\sim F(\mathbf{x}_i \,|\, \theta_i) \\ \theta_i &\sim G \\ G &\sim DP(\alpha, G_0). \end{aligned} \tag{3.1}$$

That is, the data points $\mathbf{x}_i$ are i.i.d. with distribution function $F(\mathbf{x}_i \mid \theta_i)$. The parameters θ_i specifying the data distribution are i.i.d. draws from G which is a random distribution function with a DP prior. This defines the basic DPM model.

Although the DP theory had been formally developed by Ferguson (1973), its use has become popular after the introduction of Markov chain Monte Carlo (MCMC) methods for inference. The first MCMC algorithm for DP models was introduced in the unpublished work of Escobar (1988), later published in Escobar (1994). Extensions and refinements of this method can be found in (Escobar and West, 1995; MacEachern, 1994; West et al., 1994; Bush and MacEachern, 1996). Construction of more flexible models using the DP became possible with the development of methods for the non-conjugate DPM models such as the methods in (MacEachern and Müller, 1998; Walker and Damien, 1998; Neal, 2000; Green and Richardson, 2001). These methods use a representation where the DP is integrated out. There are methods that consider approximations to DP instead of integrating it out, see for example (Muliere and Tardella, 1998; Ishwaran and James, 2002; Kottas and Gelfand, 2001). Recently, methods that use the explicit representation of the DP without having to use approximation have been developed by Papaspiliopoulos and Roberts (2005) and Walker (2006).

Other approximate inference techniques used in algorithms for DPM models include sequential importance sampling (Liu, 1996; Quintana, 1998; MacEachern et al., 1999; Ishwaran and Takahara, 2002; Fearnhead, 2004), predictive recursion (Newton and Zhang, 1999), expectation propagation (Minka and Ghahramani, 2003), and variational approximation (Blei and Jordan, 2005; Kurihara et al., 2007a,b).

Some theoretical results regarding the posterior consistency can be found in (Diaconis and Freedman, 1986; Ghosal et al., 1999). Convergence rates have been studied by (Ghosal et al., 2000; Shen and Wasserman, 2001; Walker et al., 2006).

The DPs are getting increasingly popular in machine learning. There have been several models and model extensions using the DP. Some of these include (Rasmussen, 2000; Rasmussen and Ghahramani, 2002; Beal et al., 2003; Blei et al., 2004; Dubey et al., 2004; Xing et al., 2004; Zhang et al., 2005; Daumé III and Marcu, 2005; Teh et al., 2006; Xing et al., 2006; Meeds and Osindero, 2006; Sudderth et al., 2006; Sohn and Xing, 2007; Beal and Krishnamurthy, 2006; Xue et al., 2007).

The structure of this chapter is as follows. We summarize different approaches for defining the DP distribution, and give some of the properties of the DP important for developing models and inference techniques in Section 3.1. In Section 3.2 we describe some of the MCMC algorithms to give an overview of the development of methods for inference on the DPM models. Inference techniques for the DPM models with a conjugate base distribution are relatively easier to implement than for the non-conjugate case. An important question is whether the modeling performance is weakened by using a conjugate base distribution instead of a more flexible distribution. And, a related interesting question is whether the inference is computationally cheaper for the conjugate DPM models. We address these questions in Section 3.3 using one of the most widely used DP model for density estimation and clustering: DP mixtures of Gaussians (DPMoG). The DPMoG model with both conjugate and conditionally conjugate base distributions have been used extensively in applications of the DPM models. However, the performance of the models using these different prior specifications have not been compared. We present an empirical study on the choice of the base distribution for the DPMoG model. We compare the computational cost and modeling performance of using conjugate and conditionally conjugate base distributions. When the data is believed to have a lower dimensional latent structure, it is possible to incorporate this prior knowledge to the model structure using a mixture of factor analyzers (MFA) model. In Section 3.4 we formulate the Dirichlet process mixture of factor analyzers (DPMFA) model and present experimental results on a challenging clustering problem, known as spike sorting. We conclude this chapter with a discussion in Section 3.5.

3.1 The Dirichlet Process

Let $\mathcal{X}$ be a space and $\mathcal{A}$ be a σ-field of subsets of $\mathcal{X}$. A stochastic process G on $(\mathcal{X}, \mathcal{A})$ is said to be a Dirichlet process (DP) with parameters α and G_0 if for any partition $(A_1, \dots, A_k)$, on the space of support of G_0, the random vector $(G(A_1), \dots, G(A_k))$ has a k-dimensional Dirichlet distribution[1] with parameter $(\alpha G_0(A_1), \dots, \alpha G_0(A_k))$, that is:

$$(G(A_1), \dots, G(A_k)) \sim \mathcal{D}((\alpha G_0(A_1), \dots, \alpha G_0(A_k))). \tag{3.2}$$

We denote the random probability measure G that has a DP distribution with *concentration parameter* α and *base distribution* G_0 by:

$$G \sim DP(\alpha, G_0). \tag{3.3}$$

Ferguson (1973) establishes the existence of the DP by verifying the Kolmogorov consistency conditions, appendix B.3.

Some authors define the DP using a single parameter by combining the two parameters to form the random measure $\boldsymbol{\alpha} = \alpha G_0$. Denoting the space of support of G_0 as $\mathcal{X}$, the mass of the random measure would be given as $\alpha = \boldsymbol{\alpha}(\mathcal{X})$, and the base distribution as $G_0(\cdot) = \frac{\boldsymbol{\alpha}(\cdot)}{\boldsymbol{\alpha}(\mathcal{X})}$. In the following, we use the two-parameter notation to denote the random distribution G with a DP prior, eq. (3.3).

Some important properties of the DP are as follows:

- The mean of the process is the base distribution, $\mathrm{E}\{G\} = G_0$. Thus, G_0 can be thought of as the prior guess of the shape of the distribution of G.

- Given samples $\theta_1, \dots, \theta_n$ from G, the posterior distribution is also a DP:

$$G|\theta_1, \dots, \theta_n \sim DP\Big(\alpha + n, \frac{\alpha G_0 + \sum_{i=1}^n \delta_{\theta_i}(\cdot)}{\alpha + n}\Big). \tag{3.4}$$

  Note that the concentration parameter becomes $\alpha + n$ after observing n samples, and the contribution of the prior base distribution G_0 is scaled by α. Thus, the

[1] The definition of the Dirichlet distribution and some of its properties necessary for understanding the DP are given in Appendix B.1.

concentration parameter can be seen as representing the degree of belief in the prior guess. The role of α will be discussed further in the subsequent sections.

- Draws from a DP are discrete with probability 1. As a consequence, there is positive probability of draws from a DP being identical.

Being a random probability measure, the DP can be used to express the uncertainty about the distribution of (some) parameters in a model. We can assume θ, the parameters governing the distribution of data, to have a random distribution with a DP prior:

$$\begin{aligned} \mathbf{x}_i \,|\, \theta_i &\sim F(\mathbf{x}_i \,|\, \theta_i) \\ \theta_i &\sim G \\ G &\sim DP(\alpha, G_0). \end{aligned} \tag{3.5}$$

Viewing the data distribution to be a mixture of distributions $F(\mathbf{x}_i \,|\, \theta_i)$, the measure G acts as a mixing distribution since we can write the distribution of $\mathbf{x}_i$ as

$$\mathbf{x}_i \sim \int F(\mathbf{x}_i \,|\, \theta) \mathrm{d}G(\theta). \tag{3.6}$$

Therefore, the model given in eq. (3.5) is referred to as the Dirichlet process mixture (DPM) model (Neal, 2000). The graphical representation of this model is depicted in Figure 3.1.

3.1.1 Pólya's Urn

A sequence $\{\theta_i\}_1^n, \quad (n \geq 1)$ of random variables with values in $\mathcal{X}$ is a Pólya sequence with parameters α and G_0 if for every $\theta_i \in \mathcal{X}$

$$\theta_1 \sim G_0$$

and

$$(\theta_{n+1} | \theta_1, \ldots, \theta_n) \sim G_n = \frac{\alpha G_0 + \sum_{i=1}^n \delta_{\theta_i}(\cdot)}{\alpha + n}, \tag{3.7}$$

where $\delta_\theta(\cdot)$ denotes the unit measure concentrating at θ. Imagine $\mathcal{X}$ to be the set of colors of balls in an urn, with α being the initial number of balls, and G_0 the distribution of the colors of balls such that initially there are $\alpha G_0(\theta)$ balls of color θ. The sequence $\{\theta_i\}_1^n$ described by eq. (3.7) represents the result of successive draws from the urn where after each draw, the ball drawn is replaced and another ball of the same color is added to the urn.

Blackwell and MacQueen (1973) establish the connection between the DP and Pólya sequences by extending the Pólya urn scheme to allow a continuum of colors. They show that for the extended scheme, the distribution of colors after n draws converges to a DP as $n \to \infty$. More formally, they state that if $\{\theta_i\}_1^n$ is a sequence of random variables constructed such that θ_1 has distribution G_0 and eq. (3.7) holds, then

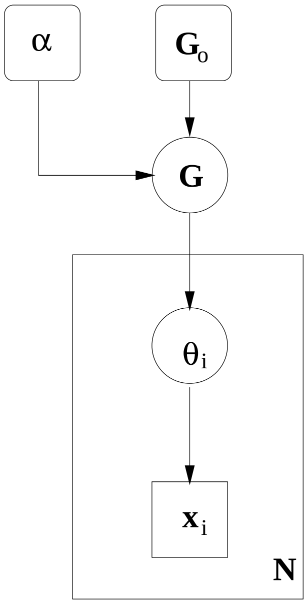

Figure 3.1: Graphical representation of a Dirichlet process mixture model. The data is assumed to be generated from a distribution parameterized by θ. The distribution of the parameter θ has a Dirichlet process prior with base distribution G_0 and concentration parameter α.

i. G_n converges almost surely as $n \to \infty$ to a random discrete distribution G.

ii. G has $DP(\alpha, G_0)$ distribution.

iii. The sequence $\{\theta_i\}_1^n$ is a sample from G.

Note that eq. (3.7) gives an expression for generating samples from G, which is infinite dimensional, without having to represent it explicitly. The graphical representation of the DPM model corresponding to this representation is depicted in Figure 3.2 .

The evolution of G_n with increasing number of samples is shown in Figure 3.3 for a Pólya urn sequence with a Gaussian base distribution and $\alpha = 5$. Note that the contribution of G_0 to the distribution of G_n gets smaller as the number of samples increases, and it vanishes for large sample sizes.

The generalized Pólya urn scheme shows that the draws from a DP exhibit a clustering property by the fact that a new sample has positive probability of being equal to one of the previous samples, and that the more often a color is sampled, the more likely it will be drawn again. Note that α determines the probability of choosing a new color. For small values of α, G_n has only a few atoms whereas for large values, the atoms are numerous, concentrating on the G_0 distribution. This is illustrated in Figure 3.3 using $\alpha = 1$ and $\alpha = 100$.

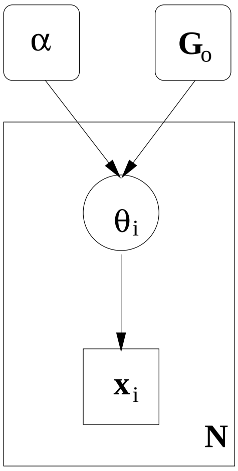

Figure 3.2: Graphical representation of the DPM using the Pólya's urn process representation. Note that G has been integrated out, and the parameters θ are drawn without referring to G with the Pólya urn scheme.

3.1.2 Chinese Restaurant Process

The Pólya urn scheme is closely related to the Chinese restaurant process (CRP) (Aldous, 1985; Pitman, 2006) which is a distribution on partitions. The CRP is a sequential process that uses the metaphor of a Chinese restaurant with an infinite number of circular tables, each with infinite seating capacity. Customers arrive sequentially at the initially empty restaurant. The first customer $\mathbf{x}_1$ sits at the first table. For $n \geq 1$, suppose n customers have already entered the restaurant and are seated in some arrangement, occupying a total of $K^{\ddagger}$ tables. Customer $\mathbf{x}_{n+1}$ chooses to sit next to customer $\mathbf{x}_l$ with equal probability $1/(n+\alpha)$ for each $1 \leq l \leq n$, and to sit alone at a new table with probability $\alpha/(n+\alpha)$. Denoting the table that customer i sits at as c_i,

$$P(c_{n+1} = k \,|\, c_1, \ldots, c_n) = \frac{\alpha}{\alpha+n}\delta_{K^{\ddagger}+1}(\cdot) + \sum_{k=1}^{K^{\ddagger}} \frac{n_k}{\alpha+n}\delta_k(\cdot), \tag{3.8}$$

where n_k denotes the number of customers seated at table k. After n customers have entered the restaurant, we have a partitioning of the customers $\{\mathbf{x}_i\}_1^n$, the partitions being defined with the variables c_i. Ignoring the labeling of the tables and focusing on only the resulting partitioning, the customers are exchangeable. That is, the order in which they enter the restaurant does not play a role in the resulting partitioning.

Suppose that independently of the sequence $\{\mathbf{x}_i\}_1^n$, we paint each occupied table by picking colors ϕ_k from the distribution over the spectrum of possible colors, G_0. Letting θ_i denote the color of the table occupied by the ith customer, the distribution of the

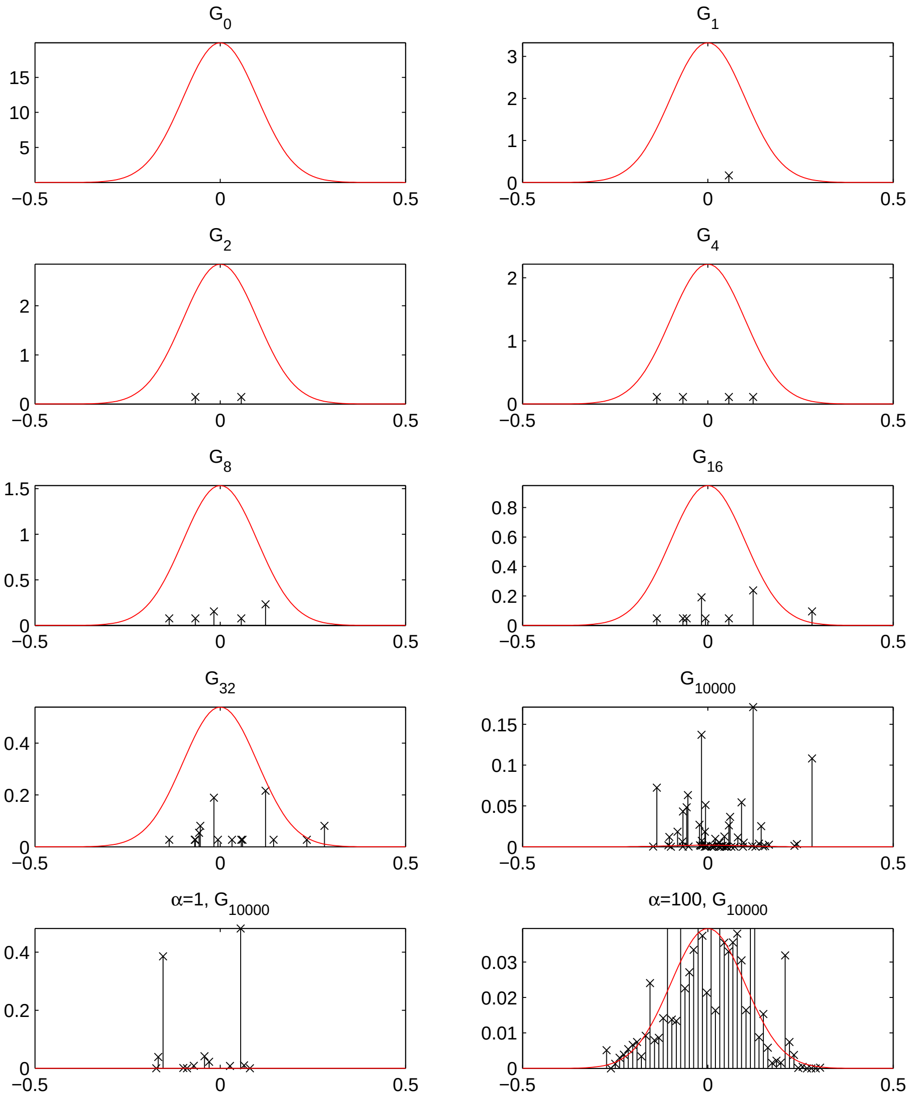

Figure 3.3: Sequential draws from the generalized Pólya urn. The base distribution G_0 is a zero-mean Gaussian with standard deviation $\sigma = 0.1$. Plots on the first four rows show the evolution of G_n with increasing sample size for the concentration parameter $\alpha = 5$. The continuous red curve shows the contribution of the base distribution, and the crosses show the atomic measures on the sampled points. For visualization, we show $\delta_\theta(\cdot)$ with a unit length line. Note that the influence of the base distribution vanishes as the sample size increases, and G_n converges to a discrete distribution. The last row shows G_n after 10000 samples for $\alpha = 1$ (left) and $\alpha = 100$ (right), showing that for large α, the draws concentrate around G_0. (The samples are binned for visualizing G_n for $\alpha = 100$.)

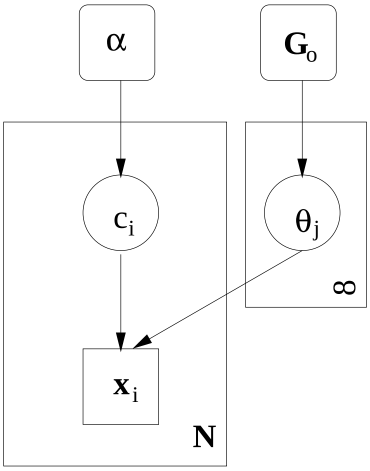

Figure 3.4: Graphical representation of the DPM model using the Chinese restaurant process. Note that the independence of the indicator variables and the parameter values are made explicit in this representation. The partitioning of the data results from the CRP, and the parameters are drawn from the base distribution G_0 independent of the partitioning.

colors would be given as

$$(\theta_{n+1}|\theta_1, \dots, \theta_n) \sim \frac{\alpha}{\alpha+n} G_0 + \sum_{k=1}^{K^\ddagger} \frac{n_k}{\alpha+n} \delta_{\phi_k}(\cdot). \tag{3.9}$$

Note that we have the same sequence of $\{\theta_i\}_1^n$ as defined by the Pólya urn scheme given in eq. (3.7). Hence, $\{\theta_i\}_1^n$ is a sample from $G \sim DP(\alpha, G_0)$.

In the CRP framework, it becomes clear that the parameter assignments (coloring of the tables) is independent from the partitioning of the data (seating arrangement). This independence is shown in the graphical model in Figure 3.4 for the DPM using the CRP representation.

3.1.3 DP as a Normalized Gamma Process

An alternative definition of the DP is given by Ferguson (1973) as a gamma process normalized to have unit mass. The intuition behind this definition follows from the definition of the Dirichlet distribution as the joint distribution of a set of independent gamma variables divided by the sum, see Appendix B.1.

A process with independent increments can be represented as a countably infinite sum of jumps of random heights at a countably infinite number of random points[2] (Ferguson and Klass, 1972). The gamma process, denoted by $Z_t \sim \Gamma\text{P}(\alpha Q(t), \beta)$, is an independent

[2] Refer to Appendix B.3 for some basic definitions about the independent increment processes.

increment process with the corresponding Lévy measure given by

$$\mathrm{d}N(x) = x^{-1}e^{-x/\beta}\alpha\mathrm{d}x, \tag{3.10}$$

where α and β are positive scalars, $t \in [0,1]$ and $Q(t)$ is a distribution function on $[0,1]$.

That is, defining the distribution of the random jump heights $J_1 \geq J_2, \ldots$ to be

$$\begin{aligned} &P(J_1 \leq x_1) = \exp\big(N(x_1)\big) \\ &P(J_j \leq x_j \,|\, J_{j-1} = x_{j-1}, \ldots, J_1 = x_1) = \exp\big(N(x_j) - N(x_{j-1})\big) \end{aligned} \tag{3.11}$$

and the random variables related to the jump times $U_1, U_2, \ldots$ to be i.i.d. Uniform$(0,1)$, independent from $J_1, J_2, \ldots$, the gamma process Z_t is defined as

$$Z_t = \sum_{j=1}^{\infty} J_j \mathbb{I}\big\{U_j \in \big[0, Q(t)\big)\big\}. \tag{3.12}$$

See Ferguson and Klass (1972) and Ferguson (1973) for details.

Let $\Gamma_{(1)} \geq \Gamma_{(2)} \geq \ldots$ denote the jump heights of a gamma process with the scale parameter $\beta = 1$. And let $\theta_k \sim G_0$ i.i.d., independent also of the $\Gamma_{(k)}$. The random measure defined as

$$G = \sum_{k=1}^{\infty} P_k \delta_{\theta_k}(\cdot) \tag{3.13}$$

has a DP(α, G_0) distribution where

$$P_k = \frac{\Gamma_{(k)}}{\sum_{j=1}^{\infty} \Gamma_{(j)}}. \tag{3.14}$$

This definition explicitly shows the discreteness of the DP as it expresses G as an infinite sum of atomic measures. The practical limitation of this construction is that we need to know the value of the infinite sum to know the value of any of the weights. The next section summarizes another infinite sum representation of the DP which does not require evaluating the infinite sum, and allows approximating the DP by truncation.

3.1.4 Stick Breaking Construction

Sethuraman and Tiwari (1982) proposed a constructive definition of the DP based on a sequence of i.i.d. random variables. The proof is provided by Sethuraman (1994). Let v_k be i.i.d. with a common distribution $v_k \sim \text{Beta}(1, \alpha)$. Define

$$\pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j), \tag{3.15}$$

and let θ_k be independent of the v_k and i.i.d. among themselves with common distribution G_0. The random probability measure G that puts weights π_k at the degenerate

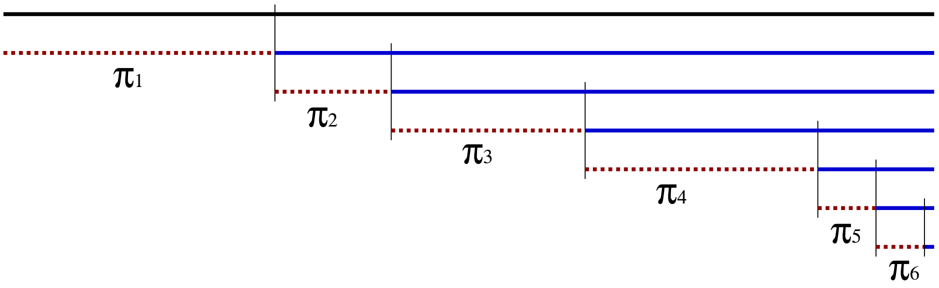

Figure 3.5: Iterative construction of the mixing proportions π_k using the stick-breaking procedure. We start with a stick of unit length (top horizontal line). The breaking points denoted with vertical lines are determined by the random variables v_k. The dotted red lines correspond to the mixing proportions π_k. These pieces are discarded after breaking off, and the breaking process is continued on the other piece shown with the blue solid lines.

measures δ_{θ_k},

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}(\cdot), \tag{3.16}$$

is distributed according to $DP(\alpha, G_0)$.

The above construction is referred to as the stick-breaking construction for the DP. The weights π_k can be imagined to be the lengths of the pieces of a unit length stick that is broken infinitely many times. Starting with a stick of unit length, at each iteration k, a piece is broken off the stick, and discarded. The breaking point is given by the random variable ν_k with $\text{Beta}(1, \alpha)$ distribution. The length of the discarded piece gives the mixing proportion π_k in the infinite mixture representation, eq. (3.16), see Figure 3.5.

The DPM using the stick-breaking representation of the DP is shown in Figure 3.6. Note that unlike the CRP or the Pólya urn representation, the random measure G is represented in this case in terms of infinitely many components with parameters θ_k and mixing proportions π_k.

Ishwaran and James (2001) show that using this construction, the DP can be approximated by truncating the number of components,

$$G_M = \sum_{k=1}^{M} \pi_k \delta_{\theta_k}(\cdot). \tag{3.17}$$

To obtain the approximation at truncation level M, we set $v_M = 1$ to have a well defined distribution.

Ishwaran and Zarepour (2000) assess the truncation accuracy by considering the behavior of the moments of the tail probability and Ishwaran and James (2002) give a bound on the truncation error, showing that the total variation between G and G_M is in the order $\exp\big(-(M-1)/\alpha\big)$. This motivates the Gibbs sampling algorithms on the truncated DP as well as variational bounds on the DP.

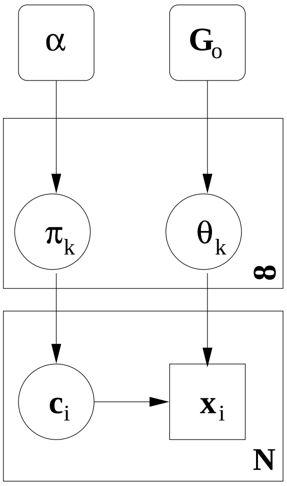

Figure 3.6: Graphical representation for the DPM using the stick-breaking construction of the DP. The DP is represented as a mixture of infinitely many atomic measures on the parameter values θ_k with mixing proportions π_k. The mixing proportions determine the prior probability of component membership for a data point $\mathbf{x}_i$.

The definition of the DP as a normalized gamma process summarized in Section 3.1.3 uses a set of random weights that are arranged in strictly decreasing order. It is interesting to note that the weights given in eq. (3.14) are equivalent to the weights π_k rearranged in decreasing order (Sethuraman, 1994; Pitman, 1996). In the stick-breaking construction, the sequence of weights is not strictly decreasing, however they are decreasing with exponential rate in expectation. The connection of the DP to the gamma process permits utilizing the truncation approach also to approximate a gamma process (Ishwaran and James, 2004).

The stick-breaking nature of the weights for the DP leads to the connection of the DP with other distributions, such as the Pitman-Yor process (Pitman and Yor, 1997) and the beta two-parameter process (Ishwaran and Zarepour, 2000), which can be seen as two-parameter extensions of the DP.

3.1.5 Infinite Mixture Models

We have seen that the DP can be defined as an infinite sum of atomic measures using the gamma process or the stick-breaking construction. These approaches show that the DPM model eq. (3.5) can be seen as a mixture model with infinitely many components. In this section, we describe another approach showing that the distribution of parameters imposed by a DP can be obtained as a limiting case of a parametric mixture model (Neal, 1992, 2000; Green and Richardson, 2001). Thereby we can gain more insight about defining DPM models as extensions of parametric models. Furthermore, this approach shows that the DP can be used as a mixture model that sidesteps the need to do model selection for determining the number of components to be used.

In the previous sections we mentioned that the parameter α expresses the strength of belief in the base distribution. Lower α values lead to only a few components being active in the model. Thus, the infinite mixture model approach suggests that α can be chosen to represent the prior expected number of active components.

The general finite mixture model with a parameter set $\Theta = \{\theta_1, \dots, \theta_K\}$ is given by:

$$P(\mathbf{x} \,|\, \Theta) = \sum_{k=1}^{K} \pi_k P(\mathbf{x} \,|\, \theta_k),$$

where π_k are the mixing proportions that are positive and sum to one, and θ_k denotes the parameters of the kth component. We place a symmetric Dirichlet distribution with parameter α/K on the mixing proportions:

$$\begin{aligned} \pi_1, \dots, \pi_K \,|\, \alpha &\sim \mathrm{Dir}(\alpha/K, \dots, \alpha/K) \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{k=1}^{K} \pi_k^{\alpha/K-1}. \end{aligned} \tag{3.18}$$

Defining indicator variables, $\mathbf{c} = \{c_1, \dots, c_n\}$, whose values encode the mixture component to which each observation belongs for the set of observations, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the mixture model can be written in the form:

$$\begin{aligned} \mathbf{x}_i \,|\, \mathbf{c}, \Theta &\sim F(\mathbf{x}_i \,|\, \theta_{c_i}) \\ c_i \,|\, \boldsymbol{\pi} &\sim \mathrm{Discrete}(\pi_1, \dots, \pi_K) \\ \theta_k &\sim G_0 \\ \boldsymbol{\pi} \,|\, \alpha &\sim \mathrm{Dir}(\alpha/K, \dots, \alpha/K). \end{aligned} \tag{3.19}$$

We can integrate over the mixing proportions $\boldsymbol{\pi}$ using the Dirichlet integral, eq. (B.11) and obtain the incremental conditional class probabilities in terms of the parameter α:

$$P(c_i = k | \mathbf{c}_{\cdot<i}, \alpha) = \frac{n_{\cdot<i,k} + \alpha/K}{i - 1 + \alpha}, \tag{3.20}$$

where the subscript $\cdot < i$ indicates all indices smaller than i, and $n_{\cdot<i,k}$ is the number of data points before $\mathbf{x}_i$ that are associated with class k.

Letting the number of mixture components K go to infinity, the conditional prior for c_i reaches the following limits:

components for which $n_{\cdot<i,k} > 0$:

$$P(c_i = k | \mathbf{c}_{\cdot<i}, \alpha) = \frac{n_{\cdot<i,k}}{i - 1 + \alpha}, \tag{3.21a}$$

any other component:

$$P(c_i \neq c_{i'} \text{ for all } i \neq i' | \mathbf{c}_{\cdot<i}, \alpha) = \frac{\alpha}{i - 1 + \alpha}. \tag{3.21b}$$

Eq. (3.21b) is the combined probability for assigning $\mathbf{x}_i$ to one of the infinitely many empty components. That is, the prior probability of assigning a data point to a component that is associated with other data points is proportional to the number of data points that have been already assigned to that component, and the probability of it having its own component is proportional to α. This approach is in accordance with the suggestion in Section 4 of (Kingman, 1975). Note the similarity of the above probabilities to eq. (3.8) and the correspondence of the indicator variables in the CRP and the infinite mixture model.

3.1.6 Properties of the Distribution

Different approaches for defining the DP summarized above reveal interesting properties of the random measure and allow development of different inference algorithms for models using DPs. Here, we summarize some properties of the DP that are important for building DPM models and developing inference techniques.

The posterior distribution given samples $\theta_1, \ldots, \theta_n$ from the process is again a DP. This property makes inference for DP models easier to handle relative to most other nonparametric models.

The Pólya urn scheme and the CRP show that it is possible to sample from the DP without having to represent the full nonparametric distribution. Thus, inference on DPM models is possible by only representing samples from the nonparametric distribution rather than the infinite dimensional distribution itself.

Draws from a DP are discrete with probability 1. As a consequence, there is positive probability of draws from a DP being identical. This property might lead to undesirable posterior properties for some models (Petrone and Raftery, 1997) but can also be exploited for defining powerful models such as the hierarchical Dirichlet process (HDP) (Teh et al., 2006) that allow sharing statistical strength between groups of data.

The DP is defined by two parameters, namely the base distribution G_0 and the concentration parameter α. The mean of the distribution is G_0, therefore G_0 can be seen as the prior distribution that we center our nonparametric model on. The concentration parameter represents the "strength of belief" on this prior distribution, similar to the scale of the parameters for the Dirichlet distribution.

The concentration parameter α controls both the smoothness (or discreteness) of the random distributions and the size of the neighborhood (or variability) of G about G_0. In the DPM models, the prior for selecting a new component to represent the ith data point is proportional to α as given in eq. (3.21b). Thus, the value of α influences the number of *active* components, that is, the number of components that have data assigned to them. The prior expected number of active components for a set of n data points is given by $\sum_{i=1}^{n} \frac{\alpha}{\alpha+i-1} \approx \alpha \log(\frac{N}{\alpha} + 1)$, (Antoniak, 1974).

For small α, the model will have a few active components drawn from the base distribution to represent the data, and for large α there will be many active components, thus the distribution of the parameters will be *concentrated* around G_0, hence the name. It is not clear what the less-informative value of α corresponds to since $\alpha \to 0$ results in representing the data with a single component and $\alpha \to \infty$ expresses strong belief that the data comes from the base distribution, (Walker et al., 1999).

The DP has been introduced as a random probability measure, and its use as a prior for modeling the uncertainty in the functional form of the distribution of parameters in a model has been motivated. In this case, larger α values pronounces stronger belief in the base distribution. The derivation of the DPM as the limit of a finite mixture model shows that it can be used as a mixture model with intrinsic selection of number of components. In this case, the concentration parameter does not necessarily need to be considered as the prior belief about the distribution of the parameters. It can be regarded as a prior guess for the number of active components in the infinite mixture model.

The likelihood for α may be derived by taking the limit $K \to \infty$ of the joint distribution for the indicator variables:

$$P(c_1, \ldots, c_n|\alpha) \propto \alpha^{K^{\ddagger}} \prod_{i=1}^{n} \frac{1}{i-1+\alpha} = \frac{\alpha^{K^{\ddagger}}\Gamma(\alpha)}{\Gamma(n+\alpha)}, \tag{3.22}$$

where $K^{\ddagger}$ denotes the number of active components. It is interesting that this expression depends only on the number of active components and the total number of data points, and not on how the observations are distributed among the components (unlike the finite mixture model case).

To allow for flexibility in the model, we need to specify a prior for α. For the simulations in this work, we chose the prior for α^{-1} to have gamma shape with unit mean and degree of freedom 1:

$$P(\alpha^{-1}) \sim \mathcal{G}(1/2, 1/2) \Longrightarrow P(\alpha) \propto \alpha^{-3/2} \exp(-1/2\alpha). \tag{3.23}$$

This prior is asymmetric, having a short tail for small values of α, expressing our prior belief, that we don't expect a very small number of active classes (say $K^{\ddagger} = 1$). Figure 3.7 shows the prior distribution on the number of components for different prior specifications on α. This figure is inspired from (Navarro et al., 2006).

The base distribution of the DP is the prior guess of the distribution of the parameters in a model. This corresponds to the distribution of the component parameters in an infinite mixture model. The shape of the priors is governed by a trade-off between modelling properties and computational considerations. In particular the use of conjugate priors makes the analysis in DPM models much more tractable, as we will see in the following section. However, the computational convenience is not a justification for using a certain type of prior that fails to represent prior knowledge. There has been a great deal of research in developing inference methods that do not require conjugacy. We describe MCMC methods for both cases in the following section, and in Section 3.3, we present an empirical study for comparing conjugate and conditionally conjugate models.

It is often difficult to directly assign the base distribution with high confidence. Thus it is customary to utilize hierarchical priors to allow sufficient flexibility of the priors when they are of parametric form (West et al., 1994; Escobar and West, 1995; Ras-

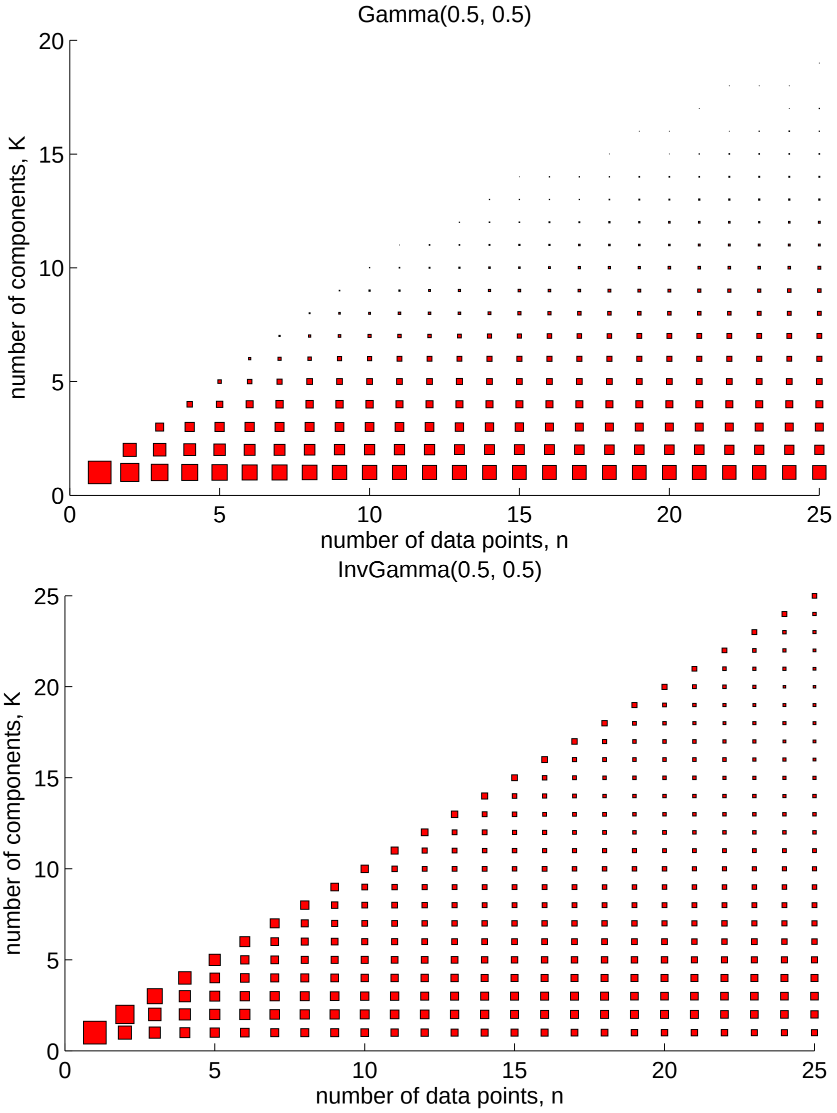

Figure 3.7: The effect of the distribution of the concentration parameter α on the prior distribution for the number of components. The plots show the change in the expected number of mixture components with increasing number of data points. The area of the squares are proportional to the probability of the corresponding number of components for a given number of data points. Note that the gamma prior favours less number of components.

mussen, 2000). In the hierarchical scheme, the priors linking the parameters of the component mixtures can all be parameterized using hyperparameters, which are themselves given vague priors. Alternatively, classical nonparametric priors such as kernel density estimation (KDE) can be used (McAuliffe et al., 2006). In this work, we consider the hierarchical model formulation.

3.2 MCMC Inference in Dirichlet Process Mixture Models

Flexible models can be obtained using a DP prior on the functional form of the parameter distributions. MacEachern and Müller (2000) show that it is possible to modify standard parametric models to obtain nonparametric models by simply using a DP over the distribution of the model parameters. The resulting model is referred to as the DPM model. Repeating from the previous section, the distributional assumptions of the DPM model are given by eq. (3.5):

$$\begin{aligned} \mathbf{x}_i \,|\, \theta_i &\sim F(\mathbf{x}_i \,|\, \theta_i) \\ \theta_i &\sim G \\ G &\sim DP(\alpha, G_0), \end{aligned} \tag{3.24}$$

where $F(\mathbf{x} \,|\, \theta)$ denotes the data distribution defined by θ. Generally, the data distribution F may depend on additional parameters, and the hierarchy may be extended by specifying priors on these parameters as well as on α and G_0. Refer to Escobar and West (1998) and MacEachern and Müller (2000) for a discussion about employing the DP prior on a part of a hierarchical model.

Analytical inference in the DPM models is not possible, therefore one has to resort to approximate techniques for inference. The MCMC methods for hierarchical models can be easily adjusted for inference on models with a nonparametric part. Similar to the hierarchical models, each parameter can be updated in turn, conditioned on the rest of the parameters and data. Since the updates for the parametric part of the model will not be affected, here we focus in the following only on inference on the nonparametric part.

Gibbs sampling for DPM models can easily be formulated and implemented when conjugate priors are used. Handling non-conjugate priors is more elaborate since the conditional posteriors cannot be analytically calculated. Nevertheless there have been several MCMC algorithms developed for inference in non-conjugate DPM models.

In the previous section, several different definitions for the DP was given. MCMC schemes that have been developed for DPM models can be divided into two classes as those that integrate out the mixing distribution G in eq. (3.24) and use the Pólya urn representation to sample the parameters θ, and those that explicitly represent G using the stick-breaking representation. The following sections give an overview of the different MCMC algorithms. We start by describing Gibbs sampling methods using the Pólya urn scheme. In Section 3.2.1 we describe the methods developed for inference on the conjugate DPM models, and in Section 3.2.2 we describe methods for the non-conjugate models. Section 3.2.3 describes the methods that explicitly represent the DP by using the stick-breaking construction.

As presented in Section 3.1, the DPM model can be seen as a mixture model with infinitely many components. For a training set of N data points, there can be at most N components that have data assigned to them, and the rest of the components will be empty. We refer to the components that have data assigned to them as the *active* components, and the empty ones as the *inactive* components. It is infeasible to represent all of the infinitely many components when doing inference. For the Pólya urn representation, the components are exchangeable, therefore we only need to explicitly represent those components that are associated with data. For the stick-breaking construction, since each component has a different stick length, the components are not exchangeable and we need to represent the components at least up to and including the last active one. Therefore for the Pólya urn representation, the set of represented components will be the same as the active components, whereas for the stick-breaking construction, the represented components might include some inactive components [3]. We refer to the components associated with only a single data point as *singleton* components.

In the following, the data points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and parameters associated with each data point $\Theta = \{\theta_1, \ldots, \theta_N\}$ are indexed by i. Due to the clustering property of the DP, some of the θ_i will be identical. Grouping the identical parameter values together as the component parameters, the number of active components will typically be much smaller than N. We use $K^\dagger$ to refer to the number of represented components, and ϕ_k, $k = 1, \ldots, K^\dagger$ to denote the parameters of each represented component. The number of active components will be denoted as $K^\ddagger$. We use indicator variables c_i that take integer values to refer to the component membership of the data points, such that $\theta_i = \phi_{c_i}$.

3.2.1 Algorithms for Conjugate Models using the Pólya Urn Scheme

In the generalized Pólya urn scheme, the distribution G of the parameters θ is integrated out. Nevertheless, the random variables obtained by the Pólya urn scheme are draws from G; see Section 3.1.1. This property can be used to do inference on the DPM models without having to explicitly represent the nonparametric part of the model given in eq. (3.24). The model can be summarized as:

$$\begin{aligned} \mathbf{x}_i \,|\, \theta_i &\sim F(\mathbf{x}_i \,|\, \theta_i) \\ \theta_i \,|\, \theta_{\cdot<i} &\sim G_n, \end{aligned} \tag{3.25}$$

where G_n is defined in eq. (3.7).

The most straightforward inference scheme defines the state of the Markov chain to be the parameters $\theta_1, \ldots, \theta_N$. Each parameter is updated from its full conditional using the Pólya urn scheme (Escobar, 1994; Escobar and West, 1995). Exploiting the exchangeability of the data points, we can assume that the parameter θ_i that we are updating belongs to the last data point, and we condition on all the other data points in the training set, and their parameters.

Treating θ_i as the last of the N samples, the prior for θ_i conditioned on all other

[3] To refer to what we call as the active and inactive components, Neal (2000) uses the terms represented and unrepresented components, and MacEachern and Müller (1998) uses the terms full and empty components, respectively. Here, we choose the terms active and inactive to have a unified terminology for the Pólya urn representation and the stick-breaking construction.

parameters θ_j for $i \neq j$, denoted as θ_{-i}, is given by the Pólya urn scheme as,

$$(\theta_i|\theta_{-i}) \sim \frac{\alpha}{\alpha+N-1}G_0(\cdot) + \frac{1}{\alpha+N-1}\sum_{j\neq i}\delta_{\theta_j}(\cdot), \tag{3.26}$$

with the mixing distribution G integrated out. Combining this with the likelihood $F(\mathbf{x}_i \,|\, \theta)$, the conditional posterior is given as

$$\begin{aligned}(\theta_i|\mathbf{x}_i, \theta_{-i}) \propto F(\mathbf{x}_i \,|\, \theta_i)\Big(\frac{\alpha}{\alpha+N-1}G_0(\cdot) + \frac{1}{\alpha+N-1}\sum_{j\neq i}\delta_{\theta_j}(\cdot)\Big)\\ =& q_{i0}G_0^i(\cdot) + \sum_{j\neq i} q_{ij}\delta_{\theta_j}(\cdot),\end{aligned} \tag{3.27}$$

where G_0^i is the posterior base line distribution based on observation $\mathbf{x}_i$ only, i.e.

$$G_0^i \propto F(\mathbf{x}_i \,|\, \theta)G_0(\theta), \tag{3.28}$$

q_{i0} is α times the marginal distribution of $\mathbf{x}_i$ under the baseline prior;

$$q_{i0} = \alpha \int F(\mathbf{x}_i \,|\, \theta)\mathrm{d}G_0(\theta), \tag{3.29}$$

and q_{ij} is the distribution of $\mathbf{x}_i$ given θ_j. The parameters are updated by repeatedly sampling from the conditional posterior given in eq. (3.27).

Algorithm 1 Gibbs sampling for conjugate DPM models with full parameter representation.

```
The state of the Markov chain consists of the parameters $\{\theta_1, \ldots, \theta_N\}$

Repeatedly sample:
for all $i = 1, \ldots, N$ do
  Update $\theta_i$ by sampling from its conditional posterior given by eq. (3.27)
end for
```

The convergence of this algorithm is expected to be poor since each parameter θ_i that is associated with a data point $\mathbf{x}_i$ is updated one at a time. The performance can be improved by making use of the clustering property of the DP, and using the representation given by the Chinese restaurant process (Section 3.1.2), or the infinite mixture models (Section 3.1.5) which demonstrate that the data points associated with identical parameters can be regarded as belonging to a component of the mixture model defined by the parameter. We can label each component and use indicator variables c_i to denote the assignment of the identical θ_i values to the component parameters ϕ_k. Instead of updating the parameter θ_i corresponding to each data point, we can update the indicator variables c_i and the component parameters ϕ_k, which would account to collectively updating all θ_i that are equal. This is the method used by West et al. (1994)

and Bush and MacEachern (1996).

In detail, the state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \ldots, c_N\}$ and the component parameters $\phi_{\mathbf{c}} = \{\phi_k, k \in \mathbf{c}\}$ The prior for the indicator variables is given by eq. (3.8). Combining this with the likelihood, the conditional posterior is;

$$\begin{aligned} &\text{assigning to components for which } n_{-i,k} > 0: \\ &\qquad P(c_i = k | \mathbf{x}_i, \mathbf{c}_{-i}, \alpha, \phi) \propto \frac{n_{-i,k}}{N - 1 + \alpha} F(\mathbf{x}_i \,|\, \phi_k), \\ &\qquad \text{assigning to a new component:} \\ &\qquad P(c_i \neq c_{i'} \text{ for all } i \neq i' | \mathbf{x}_i, \mathbf{c}_{-i}, \alpha) \propto \frac{\alpha}{N - 1 + \alpha} \int F(\mathbf{x}_i \,|\, \phi) \mathrm{d}G_0(\phi). \end{aligned} \tag{3.30}$$

That is, we either choose to assign the data point i to one of the existing components, or create a new active component by sampling its parameter from G_0^i, defined in eq. (3.28). Note that if $\mathbf{x}_i$ belongs to a singleton component, the parameter value of that component will be removed from the representation when c_i is updated. After updating the indicator variables, the component parameters are updated by sampling from their posterior conditioned on the data points assigned to them,

$$P(\phi_k \,|\, X, \mathbf{c}) \propto F(\mathbf{x}_i, \forall c_i = k \,|\, \phi_k) G_0. \tag{3.31}$$

Algorithm 2 Gibbs sampling for conjugate DPM models using indicator variables and component parameters.

```
The state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \ldots, c_N\}$ and
the component parameters $\phi_{\mathbf{c}} = \{\phi_k, k \in \mathbf{c}\}$

Repeatedly sample:
for all $i = 1, \ldots, N$ do
  Update $c_i$ using eq. (3.30)
  If $c_i$ is assigned to a singleton, sample a parameter for this component from $G_0$
end for
for all $k = 1, \ldots, K^\dagger$ do
  Update $\phi_k$ by sampling from its posterior, eq. (3.31)
end for
```

This algorithm can be further simplified by integrating over the ϕ, and eliminating them from the state (Neal, 1992; MacEachern, 1994).

The update equations for c_i is given as:

$$\begin{aligned}&\text{components for which } n_{-i,k} > 0: \\ &\qquad P(c_i = k|\mathbf{x}_i, \mathbf{c}_{-i}, \alpha) \propto \frac{n_{-i,k}}{N-1+\alpha} \int F(\mathbf{x}_i \mid \phi) G_{-i,k}(\phi) \mathrm{d}\phi, \\ &\text{another component:} \\ &\qquad P(c_i \neq c_{i'} \text{ for all } i \neq i' | \mathbf{c}_{-i}, \alpha) \propto \frac{\alpha}{N-1+\alpha} \int F(\mathbf{x}_i \mid \phi) G_0(\phi) \mathrm{d}\phi.\end{aligned} \tag{3.32}$$

where $G_{-i,k}$ is the posterior distribution obtained by updating the baseline prior with the observations assigned to component k, other than $\mathbf{x}_i$.

Algorithm 3 Gibbs sampling for conjugate DPM models using indicator variables without parameter representations.

```
The state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \dots, c_N\}$

Repeatedly sample:
for all $i = 1, \dots, N$ do
  Update $c_i$ using eq. (3.32)
end for
```

All the above methods produce ergodic Markov chains. However, they require marginalizing over the parameter θ which means that the integrals in eqs. (3.29), (3.30) and (3.32)) should be analytically tractable. This restricts the choice of the baseline prior G_0 to be conjugate to the likelihood $F(X \mid \theta)$. This requirement limits the family of DP models. West et al. (1994) suggest approximating the integral by using numerical quadrature or Monte Carlo approximation, which would provide only an approximation to the posterior. Methods for non-conjugate DP models that result in the correct stationary distribution have been developed later. MacEachern and Müller (1998) present a method that attends to the identities of the indicator variables and uses *model augmentation*. Neal (2000) follows a similar approach and uses auxiliary variables which exist only temporarily. Walker and Damien (1998) present a different auxiliary variable sampling scheme that truncates the posterior to avoid calculating the integral. In the following, we give an overview of these algorithms that do not require conjugacy, which can all be seen as extensions of the Gibbs sampling algorithm for the conjugate case. Neal (2000) also presents a combination of Metropolis Hastings proposals and Gibbs updates for the non-conjugate models, which is also summarized below. After these algorithms that use the Pólya urn representation, we describe the inference algorithms that use the stick-breaking construction.

3.2.2 Algorithms for non-Conjugate DP Models

No Gaps Algorithm

The algorithms described above use the indicator variables c_i, $i = 1, \ldots, N$ to assign identical values of θ to the component parameters ϕ. The set of numerical values of c_i do not have significance beyond denoting the grouping. The "no gaps" algorithm of MacEachern and Müller (1998) constrains the labels of the active components to cover the integers from 1 to $K^\ddagger$, and augments the state to include empty components so as to have a total of N represented components, i.e. $K^\dagger = N$. Augmenting the model replaces the integral evaluations with likelihood evaluations.

The state of the Markov chain consists of the indicator variables and the parameters of the N components, $K^\ddagger$ of which have data assigned, the rest being empty. First the indicator variable of each data point i is updated as follows. Use $K^\ddagger_-$ to denote the number of distinct groups of data not including the ith data point. Label the groups from 1 to $K^\ddagger_-$. If c_i is a singleton, with probability $K^\ddagger_-/(K^\ddagger_- + 1)$ leave c_i unchanged, otherwise label c_i as $(K^\ddagger_- + 1)$, consequently assigning $\phi_{K^\ddagger_- +1}$ to be the existing value for ϕ_{c_i}. Update c_i with the following probabilities:

$$\begin{aligned} P(c_i = k \,|\, \mathbf{x}_i, c_{-i}, \phi) &\propto n_{-i,k} F(\mathbf{x}_i \,|\, \phi_k) \quad \text{for } k = 1, \ldots, K^\ddagger_-, \\ P(c_i = K^\ddagger_- + 1 \,|\, \mathbf{x}_i, c_{-i}, \phi) &\propto \frac{\alpha}{K^\ddagger_- + 1} F(\mathbf{x}_i \,|\, \phi_{K^\ddagger_- +1}) \end{aligned} \tag{3.33}$$

After the indicator updates, update the component parameters by sampling from their conditional posterior, eq. (3.31).

Algorithm 4 The No-Gaps algorithm for non-conjugate DPM models.

```
The state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \ldots, c_N\}$ and
$N$ component parameters $\Phi = \{\phi_1, \ldots, \phi_N\}$
Of the $N$ parameters, only $K^\ddagger + 1$ of them are represented

Repeatedly sample:
for all $i = 1, \ldots, N$ do {indicator updates}
  Let $K^\ddagger_-$ denote the number of active components without considering $\mathbf{x}_i$
  if $c_i$ is a singleton then
    With probability $K^\ddagger_-/(K^\ddagger_- + 1)$ do not update $c_i$,
    otherwise, label $c_i$ as $K^\ddagger_- + 1$
  end if
  Label the components of all data points other than $i$ from 1 to $K^\ddagger_-$
  Update $c_i$ using eq. (3.33)
end for
for all $k = 1, \ldots, N$ do {parameter updates}
  Update $\phi_k$ by sampling from its posterior, eq. (3.31)
end for
```

The no gaps algorithm augments the state to have N components, some of which will be empty. The values of $\phi_1, \ldots, \phi_N$ are not altered during indicator variable updates. Only group labels are changed if needed and the component parameters are updated only after updating the indicators of all data points. Note that although theoretically the state includes N components, we only need to represent $K_-^\ddagger + 1$ of them since the rest of the components are not considered when updating c_i. We can include new components in the representation by sampling new parameter values from the prior as needed.

Gibbs Sampling Using Auxiliary Variables (Neal, 2000)

Neal (2000) presents another Gibbs sampling algorithm that uses auxiliary variables that only exist temporarily, instead of augmenting the state to include empty components.

The state of the Markov chain consists of the indicator variables $c_1, \ldots, c_N$ and the active component parameters ϕ_c. We repeatedly update the indicator variables and the component parameters. The component parameters can be updated by simply sampling from their posterior. When we update the indicator variable for a particular data point, $\zeta \geq 1$ auxiliary components will be used to avoid the intractable integral.

Treating $\mathbf{x}_i$ as the last data point, we can either assign c_i to be equal to one of the active components, or create a new component. The auxiliary components will represent the possible new components. Using the same notation as above, there are $K_-^\ddagger$ active components associated with the data points other than i. The prior probability of assigning c_i to be equal to one of the active components is $n_{-i,k}/(N-1+\alpha)$ and the probability of creating a new component is $\alpha/(N-1+\alpha)$, which will be split among the ζ auxiliary components.

If c_i is a singleton, then we assign the parameter of one of the auxiliary variable to be ϕ_{c_i}, and draw values for the parameters of the rest of the auxiliary variables from G_0. Otherwise, if c_i belongs to a component that is also associated with other data points, then we sample parameters of all auxiliary components form G_0. We update c_i using the following probabilities:

$$\begin{aligned} P(c_i = k \mid \mathbf{x}_i, c_{-i}, \phi) &\propto \frac{n_{-i,k}}{N-1+\alpha} F(\mathbf{x}_i \mid \phi_k) && \text{for } k = 1, \ldots, K_-^\ddagger, \\ P(c_i = k \mid \mathbf{x}_i, c_{-i}, \phi) &\propto \frac{\alpha/\zeta}{N-1+\alpha} F(\mathbf{x}_i \mid \phi_k) && \text{for } k = (K_-^\ddagger + 1), \ldots, (K_-^\ddagger + \zeta). \end{aligned} \tag{3.34}$$

Note the differences between this method and the no gaps algorithm. Here, we have used $k = 1, \ldots, K_-^\ddagger$ as the labels for the active components, and $k = K_-^\ddagger + 1, \ldots, K_-^\ddagger + \zeta$ for the auxiliary components only for notational convenience. On the other hand, the values of the labels were significant for the no gaps algorithm of the previous section. Furthermore, the prior probability of creating a new component is $K_-^\ddagger + 1$ greater for this algorithm. For large values of ζ, the auxiliary variables can be thought of approximating G_0 and giving an approximation of the integral in eq. (3.29). But the algorithm is valid for any $\zeta \geq 1$. ζ can be chosen as a compromise between computational cost and mixing performance.

Algorithm 5 Gibbs sampling for non-conjugate DPM models using auxiliary components

```
The state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \ldots, c_N\}$ and
the parameters of the active components $\Phi = \{\phi_1, \ldots, \phi_{K^\ddagger}\}$

Repeatedly sample:
for all $i = 1, \ldots, N$ do {indicator updates}
  if $c_i$ is a singleton then
    Assign $\phi_{c_i}$ to be the parameter of one of the auxiliary components
    Draw values from $G_0$ for the rest of the auxiliary parameters
  else
    Draw values from $G_0$ for all the $\zeta$ auxiliary parameters
  end if
  Update $c_i$ using eq. (3.34)
  Discard the inactive components
end for
for all $k = 1, \ldots, K^\ddagger$ do {parameter updates}
  Update $\phi_k$ by sampling from its posterior, eq. (3.31)
end for
```

Metropolis-Hastings Updates

Neal (2000) proposes to use Metropolis-Hastings updates combined with *partial* Gibbs updates. We can use a Metropolis Hastings algorithm using the conditional priors as the proposal distribution, leading to the acceptance probability to be the ratio of the likelihoods. That is, we propose c_i to be equal to one of the existing components with probability $n_{-i,k}/(N-1+\alpha)$ and propose to create a singleton with probability $\alpha/(N-1+\alpha)$. The proposal is evaluated with the ratio of the likelihoods.

The probability of considering to create a new component would be very low if α is small relative to the number of data points N. Therefore Neal (2000) changes the proposal distribution so as to increase the probability of considering to propose forming a new component. And, to facilitate mixing, he adds partial Gibbs sampling steps for the members of the non-singleton components in which only changing c_i to one of the existing components is considered. In detail, if the data point i belongs to a singleton, it will be considered to be assigned to only one of the existing components with assignment probability $n_{-i,k}/(N-1)$. The acceptance ratio of this proposal is

$$\min\left\{1, \frac{\alpha}{N-1}\frac{F(\mathbf{x}_i \,|\, \phi_{c_i^*})}{F(\mathbf{x}_i \,|\, \phi_{c_i})}\right\}. \tag{3.35}$$

And, whenever c_i is not a singleton, proposing to change it to a newly created component will be proposed with probability 1, and a parameter for this component will be sampled from the prior G_0, resulting in the acceptance ratio:

$$\min\left\{1, \frac{N-1}{\alpha}\frac{F(\mathbf{x}_i\,|\,\phi_{c_i^*})}{F(\mathbf{x}_i\,|\,\phi_{c_i})}\right\}. \tag{3.36}$$

Algorithm 6 Metropolis-Hastings sampling with restricted Gibbs updates for non-conjugate DPM models

```
The state of the Markov chain consists of the indicator variables $\mathbf{c} = \{c_1, \dots, c_N\}$ and
the parameters of the active components $\Phi = \{\phi_1, \dots, \phi_{K^\ddagger}\}$

Repeatedly sample:
for all $i = 1, \dots, N$ do
  if $c_i$ is a singleton then
    Assign $c_i$ to one of the active components with probability $\frac{n_{-i,k}}{N-1}$
    Evaluate the assignment with acceptance probability given in eq. (3.35)
  end if
  if $c_i$ is not a singleton then
    Assign $c_i$ to a new component drawing a parameter $\phi_{c_i}$ from $G_0$
    Evaluate the assignment with acceptance probability given in eq. (3.36)
  end if
end for
for all $c_i$ that are not singletons do
  Assigning $c_i$ to one of the active components with probability $\propto \frac{n_{-i,k}}{N-1}F(\phi_k)$
end for
for all $k = 1, \dots, K^\ddagger$ do
  Update $\phi_k$ by sampling from its conditional posterior, eq. (3.31)
end for
```

Gibbs Sampling using Auxiliary Variables (Walker and Damien, 1998)

In this section, we summarize another Gibbs sampling algorithm using auxiliary variables, which is rather different from the methods discussed so far.

Repeating from Section 3.2.1, we want to sample from the conditional posterior of the parameter given by eq. (3.27),

$$\begin{aligned}(\theta_i|\mathbf{x}_i, \theta_{-i}) \propto& F(\mathbf{x}_i\,|\,\theta_i)\Big(\frac{\alpha}{\alpha+n-1}G_0(\cdot) + \frac{1}{\alpha+n-1}\sum_{j\neq i}\delta_{\theta_j}(\cdot)\Big)\\ =& q_{i0}G_0^i(\cdot) + \sum_{j\neq i} q_{ij}\delta_{\theta_j}(\cdot),\end{aligned} \tag{3.37}$$

which requires computing the integral $q_{i0} = \alpha\int F(\mathbf{x}|\theta)\mathrm{d}G_0(\theta)$. To avoid this integral which is intractable for non-conjugate models, Walker and Damien (1998) suggest introducing a latent variable u_i and construct the joint distribution of (θ_i, u_i) given by

$$(\theta_i, u_i \,|\, \mathbf{x}_i) \propto \mathbb{I}\{u < F(\mathbf{x}_i \,|\, \theta_i)\}\big(\alpha G_0(\theta_i) + \sum_{j \neq i} \delta_{\theta_j}(\theta_j)\big), \tag{3.38}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. The full conditionals for the Gibbs sampling are given as

$$(u_i|\theta_i, \mathbf{x}_i) \sim \text{Uniform}\big(0, F(\mathbf{x}_i \,|\, \theta_i)\big) \tag{3.39}$$

and

$$(\theta_i|u_i, \mathbf{x}_i) \propto \alpha G_0(\theta_i)\mathbb{I}\{u < F(\mathbf{x}_i \,|\, \theta_i)\} + \sum_{j: u < F(\mathbf{x}_i \,|\, \theta_i)} \delta_{\theta_j}(\theta_j) \tag{3.40}$$

The above equation constrains the space of sampling by using the auxiliary variable u. We set θ_i to be equal to θ_j with probability proportional to 1 if the likelihood of θ_j is greater than u, and with probability proportional to α, we sample from G_0 restricted to the range where $u < F(\mathbf{x}_i \,|\, \theta_i)$. Thus, we avoid evaluating the integral. Note that, this method is applicable only to the models in which it is feasible to sample from the truncated form of G_0.

Algorithm 7 Auxiliary variable sampling for non-conjugate DPM models.

```
The state of the Markov chain consists of the parameters $\Theta = \{\theta_1, \ldots, \theta_N\}$

Repeatedly sample:
for all $i = 1, \ldots, N$ do
    Sample an auxiliary variable $u_i$ with conditional distribution given by eq. (3.39)
    Update $\theta_i$ by sampling from the truncated posterior given the auxiliary variable, eq. (3.40)
    Discard the auxiliary variable
end for
```

Above, we have summarized several MCMC algorithms for inference on the DPM models using the Pólya urn representation. Using this representation, the cluster labels are exchangeable and therefore the distribution does not depend on the cluster identities. A common property of the presented methods is that they are all based on incremental updates. That is, either the parameter or the indicator variable associated with each data point is updated incrementally. The mixing behavior of the samplers can be improved by using more complex sampling steps that can escape local modes. Green and Richardson (2001) describe a split-merge sampler using RJMCMC (Green, 1995), which may be difficult to construct for multivariate data. As an easier to formulate alternative, Jain and Neal (2000) introduce a split-merge sampler using a Metropolis-Hastings procedure for conjugate DPM models. A similar sampler that can work in the nonconjugate case is introduced in Jain and Neal (2005). These split-merge methods can be combined with any of the above methods to obtain a better mixing chain.

3.2.3 Algorithms Using the Stick-Breaking Representation

The previous sections presented sampling schemes that do not explicitly represent the random distribution G with a DP prior, and only work on samples from G. The alternative to this approach is to explicitly represent the DP using the stick-breaking construction, and update it as well as the parameters drawn from it. The equivalent model can be written as:

$$\begin{aligned}
\mathbf{x}_i \,|\, \theta_i &\sim F(\mathbf{x}_i \,|\, \theta_{c_i}) \\
c_i &\sim \sum_{k=1}^{\infty} \pi_k \delta_k(\cdot) \\
v_k &\sim \text{Beta}(1, \alpha), \quad \pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j) \\
\theta_k &\sim G_0
\end{aligned} \tag{3.41}$$

In this formulation, we need to represent the mixing proportions (stick lengths) and the parameters of each mixture component as well as the indicator variables denoting which component each observation is assigned to.

Gibbs Sampling for the Truncated DP

In the stick-breaking construction of the DP, the mixing proportions π_k decrease exponentially fast in expectation. This motivates using a truncation of DP which would be easier to handle than the infinite case. Ishwaran and James (2001) give a bound for the error introduced by truncating the DP, showing that truncating the number of mixture components at a moderate level is sufficient to successfully approximate the DP, see Section 3.1.4. Gibbs sampling using the truncated DP is straightforward since it accounts to inference on a finite mixture model.

Truncating the number of mixtures in eq. (3.41) to M, the state consists of the component parameters $\theta_1, \dots, \theta_M$, the mixing proportions $\pi_1, \dots, \pi_M$ and the indicator variables $c_i, \dots, c_N$ which we repeatedly update. The conditional posterior for the component parameters is as given in eq. (3.31) in the previous sections. The conditional posterior for the mixing proportions again have a stick-breaking construction given as

$$\pi_1 = v_1^*, \text{ and } \pi_k = v_k^* \prod_{l=1}^{k-1} v_l^*, \tag{3.42}$$

where

$$v_k^* \sim \text{Beta}(1 + n_k, \alpha + \sum_{l=k+1}^{M} n_l). \tag{3.43}$$

Since the assignments of the data points are independent given the mixing proportions, we can update the indicator variables jointly, without conditioning on the other indicator variables. This might result in faster mixing chains compared to the Pólya urn samplers in some cases. The conditional for c_i is

$$c_i \,|\, \mathbf{x}_i, \pi_1, \ldots, \pi_M, \theta_1, \ldots, \theta_M \;\sim\; \sum_{k=1}^{M} \pi_{ki} \delta_k(\cdot) \tag{3.44}$$

where the mixing proportions of the posterior are given by $\pi_{ki} \propto \pi_k F(\mathbf{x}_i \,|\, \theta_k)$.

Algorithm 8 Gibbs sampling for truncated DP

```
The state of the Markov chain consists of the component parameters $\theta_1, \ldots, \theta_M$,
mixing proportions $\pi_1, \ldots, \pi_M$ and indicator variables $c_i, \ldots, c_N$.

Repeatedly sample:
for all $i = 1, \ldots, N$ do {indicator updates}
  Assign $c_i$ to one of the $M$ components with probability $\propto \pi_k F(\mathbf{x}_i \,|\, \theta_k)$
end for
for all $k = 1, \ldots, M$ do {parameter updates}
  Update $\theta_k$ by sampling from its posterior $\propto F(\mathbf{x}_i, \forall c_i = k \,|\, \theta_k) G_0$
end for
for all $k = 1, \ldots, M$ do {mixing proportion updates }
  Sample the posterior breaking points $v_k^*$ using eq. (3.43)
  Set $\pi_k = v_k^* \prod_{l=1}^{k-1} v_l^*$
end for
```

This algorithm does not require conjugacy, as we do not need to integrate over the parameters for indicator variable updates. Gibbs sampling is easy to implement for the truncated Dirichlet process using the stick-breaking construction. However, it is desirable to avoid approximations and sample from the exact posterior distribution. In the following sections, we describe methods by Papaspiliopoulos and Roberts (2005) and Walker (2006) that use the stick-breaking representation without truncating the process.

Retrospective Sampling

Retrospective sampling of Papaspiliopoulos and Roberts (2005) suggests allocating components as needed instead of using truncation. Since the stick lengths sum up to 1, prior assignment of the indicator variables is straightforward by starting with only a few breaks of the stick (components) and breaking the stick more as a point in the unbroken part is sampled, see Figure 3.8. The posterior distribution of the indicator variables is given by the combination of the mixing proportions and the likelihood. In this representation, the clusters are not exchangeable since the mixing proportions differ. Therefore, we would need to sum over the infinitely many cases to obtain the normalizing constant for the posterior. Retrospective sampling provides a way to avoid evaluating this infinite sum by using a Metropolis-Hastings step.

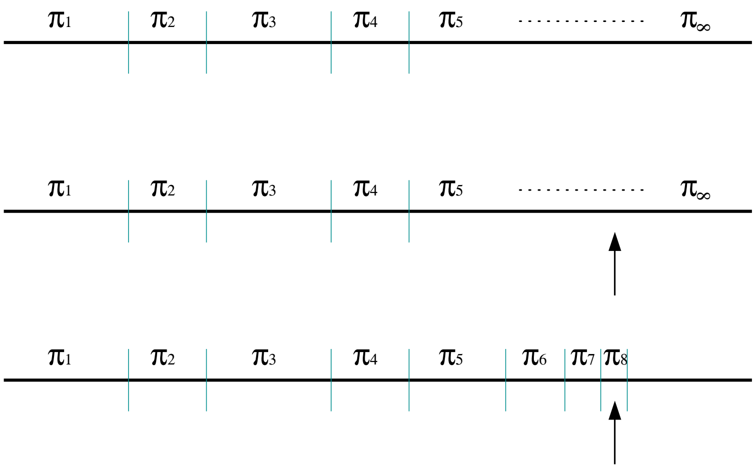

Figure 3.8: Illustration of extending the DP representation retrospectively. The black horizontal line is a stick of unit length. The blue vertical lines show the breaking points. The length of each broken piece corresponds to a mixing proportion. Initially there are only four components represented (top). To decide on the assignment of the data point, a value is sampled uniformly from [0, 1], shown by the arrow (middle). If the part that the arrow points at is not already represented, more pieces are represented retrospectively by breaking the stick and sampling parameter values for the new pieces until the arrow falls on a represented piece (bottom).

At each iteration, we denote the last component that has data assigned to it as $K^\dagger$. We only represent the parameters and the mixing proportions of components with index $k \leq K^\dagger$. We can assign a data point either to one of the represented components or to one of the rest. The posterior for the indicator variable c_i is given as

$$\begin{aligned} P(c_i = k \,|\, \mathbf{x}_i) &\propto \pi_k F(\mathbf{x}_i \,|\, \theta_k) \quad \text{for } k \leq K^\dagger \\ P(c_i = k \,|\, \mathbf{x}_i) &\propto \pi_k M_i \quad \text{for } k > K^\dagger \end{aligned} \tag{3.45}$$

with normalizing constant

$$\kappa(K^\dagger) = \sum_{k=1}^{K^\dagger} \pi_k F(\mathbf{x}_i \,|\, \theta_k) + (1 - \sum_{k=1}^{K^\dagger} \pi_k) M_i, \tag{3.46}$$

where M_i is a constant controlling the probability of proposing an assignment to one of the unrepresented components. Papaspiliopoulos and Roberts (2005) choose M_i such that posterior probability of allocating i to a new component is greater than the prior. With the choice of $M_i(K^\dagger) = \max_{k \leq K^\dagger} F(\mathbf{x}_i \,|\, \theta_k)$, the Metropolis-Hastings acceptance ratio of changing the component assignment c_i from k to j is given as:

$$\alpha = \begin{cases} 1, & \text{if } j \leq K^\dagger \text{ and } K^\dagger = K^\dagger_{kj} \\ \min\left\{1, \dfrac{\kappa(K^\dagger)}{\kappa(K^\dagger_{kj})} \dfrac{M(K^\dagger_{kj})}{F(\mathbf{x}_i \,|\, \theta_k)}\right\}, & \text{if } j \leq K^\dagger \text{ and } K^\dagger_{kj} < K^\dagger \\ \\ \min\left\{1, \dfrac{\kappa(K^\dagger)}{\kappa(K^\dagger_{kj})} \dfrac{F(\mathbf{x}_i \,|\, \theta_j)}{M(K^\dagger)}\right\}, & \text{if } j > K^\dagger \end{cases} \tag{3.47}$$

where we used $K^\dagger_{kj}$ to denote the index of the last active component after changing component assignment of $\mathbf{x}_i$ form k to j. Thus, the algorithm behaves like a Gibbs sampler when the assignment to an already represented component is proposed if this move does not change last active component. This algorithm provides a sampler that does not need to resort to approximations. However, the choice of M_i should be made carefully to satisfy detailed balance when the index for the last active component changes and obtain a good mixing performance.

Algorithm 9 Retrospective sampling for DPM models using stick-breaking construction

```
The state of the Markov chain consists of the stick lengths (mixing proportions) and
the component parameters.
Only the mixing proportions and parameters of the components up to and including
the last active component are represented.

Repeatedly sample:
for all $i = 1, \ldots, N$ do
  Sample $u_i \sim \text{Uniform}[0, 1]$
  Evaluate the posterior probabilities for component assignments using eq. (3.45).
  if $\sum_{l=1}^{k} P(c_i = l) > u_i$ for $k \leq K^\dagger$ then
    Represent more components by breaking the stick to obtain the mixing propor-
    tions and sampling parameters from the prior until $\sum_{l=1}^{k} P(c_i = l) < u_i$ for
    $l \leq K^\dagger$.
  end if
  Propose to move to $c_i = \inf\{k \leq K^\dagger : \sum_{l=1}^{k} P(c_i = l) < u_i\}$
  Evaluate the proposal with eq. (3.47)
end for
```

Slice Sampling

The idea of auxiliary variable methods is sampling for the variables we are interested in by using auxiliary variables which make the updates easier to handle. Slice sampling is an auxiliary variable method that exploits the fact that sampling from a distribution is equivalent to sampling uniformly from the region under its probability density function

(Neal, 2003). Thus, the problem of sampling from an arbitrary distribution reduces to sampling from uniform distributions.

In Section 3.2.2 we have described an algorithm by Walker and Damien (1998) that uses auxiliary variables to limit the space of sampling in the Pólya urn representation. The auxiliary variable in that algorithm is chosen such that it has uniform distribution defined by the likelihood value. Therefore, given the auxiliary variable, sampling from the posterior reduces to sampling from a truncated version of the prior.

In this section, we describe a similar idea applied to the stick-breaking construction of the DP by Walker (2006) that results in an elegant algorithm which is widely applicable. The parameters, the mixing proportions and the indicator variables are repeatedly updated. We introduce the temporary slice variable s when updating the indicators, and discard it after the indicator update.

The distribution of the auxiliary variable s is defined such that the joint prior of s and c_i is a two-dimensional uniform distribution. Conditioning on s, c_i is uniformly distributed on a limited part of the prior space. Combining this with the likelihood, we have the conditional posterior of the c_i.

Recall that the prior probability of assigning an observation to one of the components is given by the mixing proportions $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_\infty\}$,

$$P(c_i|\boldsymbol{\pi}) = \pi_{c_i}. \tag{3.48}$$

Multiplying with the likelihood, the posterior is,

$$P(c_i|\boldsymbol{\pi}, \mathbf{x}_i, \theta) \propto \pi_{c_i} F(\mathbf{x}_i|\theta_{c_i}). \tag{3.49}$$

We introduce the auxiliary slice variable s such that the joint posterior of the indicator variable and s is

$$P(c_i, s|\boldsymbol{\pi}, \mathbf{x}_i, \theta) \propto \mathbb{I}\{s < \pi_{c_i}\} F(\mathbf{x}_i|\theta_{c_i}). \tag{3.50}$$

Thus, the distribution of s given $\boldsymbol{\pi}$ and c_i is uniform:

$$(s|\boldsymbol{\pi}, c_i) \sim \mathrm{U}(0, \pi_{c_i}) = \mathbb{I}\{s < \pi_{c_i}\}\pi_{c_i}^{-1} \tag{3.51}$$

and the distribution of c_i conditioned also on s is

$$P(c_i|s, \boldsymbol{\pi}, \mathbf{x}_i, \theta) \propto \begin{cases} F(\mathbf{x}_i|\theta_{c_i}) & \text{if } s < \pi_{c_i}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.52}$$

That is, the probability of assigning c_i to components with mixing proportions less than the slice variable is 0. Therefore, we only need to consider assignment to one of the components that have a larger stick length than the slice variable s. This will clearly be a finite number, rather than the infinitely many components.

Using slice sampling, we only need to represent the mixing proportions and the parameters of the $K^\dagger$ components. We allocate new components only when needed. The slice value is sampled uniformly between 0 and π_{c_i}. Note that the stick lengths are not

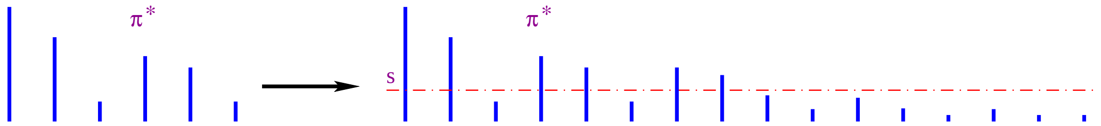

Figure 3.9: Slice sampling for DP. The vertical lines denote the mixing proportions π_k of the represented components in the stick breaking construction. Initially, there are six components represented and for the data point being considered, $c_i = 4$ (left). A slice value s (shown by the dotted red line) is sampled from Uniform$[0, \pi_4]$ and the represented components are extended until the sum of the remaining mixing proportions is less than s (right).

strictly decreasing, but we know that they have to sum to 1. Therefore, if $K^\dagger$ components are represented, one of the unrepresented components can have a maximum stick length of $1 - \sum_{k=1}^{K^\dagger} \pi_k$. If this value is greater than s, we keep breaking the stick until we are left with a stick length smaller than s, see Figure 3.9 for a pictorial representation. We sample parameters for the newly allocated components from the prior. Then we update the indicator variable given the data and the components above the slice.

Algorithm 10 Slice sampling algorithm for the DPM model using the stick-breaking construction.

The state of the Markov chain consists of indicator variables for each data point and the infinitely many components with corresponding mixing proportions and parameters.

Only the mixing proportions and parameters of the components up to and including the last active component are represented.

```
Repeatedly sample:
for all $i = 1, \dots, N$ do {indicator updates}
  Sample a slice variable $s$ from the conditional distribution eq. (3.51)
  if $s < \sum_{l=1}^{K^\dagger} \pi_l$ then
    Extend the representation by breaking the stick until $s > \sum_{l=1}^{K^\dagger} \pi_l$
    Sample parameters for the new represented stick pieces from $G_0$
  end if
  Assign $\mathbf{x}_i$ to one of the components above the slice, using eq. (3.52)
end for
```

The methods presented in this section use the stick-breaking construction of the DP. In this construction, the mixing proportions of the (infinitely many) mixture components are represented. Therefore the indicator variables are updated without conditioning on the other indicators. This feature of the samplers encourages good mixing for the indicator variables. However, it should be noted that since the mixing proportions of each component is represented explicitly, with a size-biased ordering of the cluster labels, the components are not exchangeable in this representation. Therefore, caution should be taken for mixing over the cluster labels to avoid clustering bias. Porteus et al. (2006) discuss this in detail and introduce moves that permutes or swaps the labels to improve mixing over the cluster labels.

3.3 Empirical Study on the Choice of the Base Distribution

In the previous section, we have described several different algorithms for doing inference on the DPM models with both conjugate and non-conjugate base distribution. We have seen that inference in the conjugate DPM models is relatively straightforward. For some models, it is not possible to specify priors conjugate to the likelihood. The question is whether one should use conjugate priors at all when they are available. It is known that generally, conjugacy limits the flexibility of the models. Is the computational ease worth the price of worse modeling performance? Or, is using conjugate models really computationally cheaper than the non-conjugate alternatives, and how does the modeling performance change with the choice of priors? In this section, we seek to empirically address these questions using a DP mixture of Gaussians model (DPMoG). We choose to use a DPMoG because it is one of the most widely used DPM models for which it is possible to employ a conjugate and a conditionally conjugate base distribution.

In the following, we give model formulations for both a conjugate and a conditionally-conjugate base distribution. For both prior specifications, we define hyperpriors on G_0 for robustness. We refer to the models with the conjugate and the conditionally conjugate base distributions in short as the conjugate model and the conditionally conjugate model, respectively. After specifying the model structure, we will discuss in detail how to do inference on both models. We will show that mixing performance of the non-conjugate sampler can be improved substantially by exploiting the conditional conjugacy. We will present experimental results comparing the modeling performance of the two models and the computational cost of the samplers on several data sets.

3.3.1 The Dirichlet Process Gaussian Mixture Model

The finite Gaussian mixture model may be written as:

$$p(\mathbf{x}_i|\theta_1,\dots,\theta_K) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, S_j^{-1}) \tag{3.53}$$

where $\theta_j = \{\boldsymbol{\mu}_j, S_j\}$ is the set of parameters for component j, π_j are the *mixing proportions* (which must be positive and sum to one), $\boldsymbol{\mu}_j$ is the mean vector for component j, and S_j is the component precision (inverse covariance matrix). Defining a joint prior distribution G_0 on the component parameters and introducing indicator variables, the model can be written in the form of eq. (3.19):

$$\begin{aligned}
\mathbf{x}_i \,|\, c_i, \Theta &\sim \mathcal{N}(\boldsymbol{\mu}_{c_i}, S_{c_i}^{-1}) \\
c_i \,|\, \boldsymbol{\pi} &\sim \text{Discrete}(\pi_1, \ldots, \pi_K) \\
(\boldsymbol{\mu}_j, S_j) &\sim G_0 \\
\boldsymbol{\pi} \,|\, \alpha &\sim \mathcal{D}(\alpha/K, \ldots, \alpha/K).
\end{aligned} \tag{3.54}$$

We obtain the Dirichlet Process mixture of Gaussians (DPMoG) model by integrating out the mixing proportions and taking the limit $K \to \infty$, as discussed in Section 3.1.5. Equivalently, we can define the model by starting with Gaussian distributed data

$$\mathbf{x}_i | \theta \sim \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_i, S_i^{-1}), \tag{3.55}$$

with a random distribution of the model parameters $(\boldsymbol{\mu}_i, S_i) \sim G$ drawn from a DP, $G \sim DP(\alpha, G_0)$.

We put an inverse gamma prior on the DP concentration parameter α,

$$\alpha^{-1} \sim \mathcal{G}(1/2, 1/2), \tag{3.56}$$

the choice of which is discussed in detail in Section 3.1.6. We need to specify the base distribution G_0 to complete the model. For DPMoG, G_0 specifies the mean of the joint distribution of $\boldsymbol{\mu}$ and S. For this model, a conjugate base distribution exists however it has the unappealing property of prior dependency between the mean and the covariance. We proceed by giving the detailed model formulation for both conjugate and conditionally conjugate cases.

Conjugate DPMoG

The natural choice of priors for the mean of the Gaussian is a Gaussian, and a Wishart distribution for the precision (inverse-Wishart for the covariance). To accomplish conjugacy of the joint prior distribution of the mean and the precision to the likelihood, the distribution of the mean has to depend on the precision. The prior distribution of the mean $\boldsymbol{\mu}_j$ is Gaussian conditioned on S_j:

$$(\boldsymbol{\mu}_j | S_j, \boldsymbol{\xi}, \rho) \sim \mathcal{N}\big(\boldsymbol{\xi}, (\rho S_j)^{-1}\big), \tag{3.57}$$

and the prior distribution of S_j is Wishart:

$$(S_j | \beta, W) \sim \mathcal{W}\big(\beta, (\beta W)^{-1}\big). \tag{3.58}$$

The joint distribution of $\boldsymbol{\mu}_j$ and S_j is the Normal/Wishart distribution denoted as:

$$(\boldsymbol{\mu}_j, S_j) \sim \mathcal{NW}(\boldsymbol{\xi}, \rho, \beta, \beta W), \tag{3.59}$$

with $\boldsymbol{\xi}, \rho, \beta$ and W being hyperparameters common to all mixture components, expressing the belief that the component parameters should be similar, centered around some

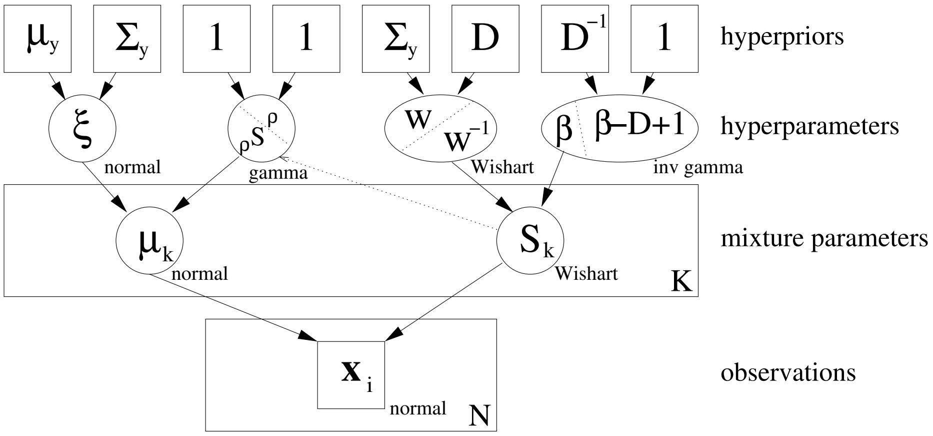

Figure 3.10: Graphical representation of the layered structure of the hierarchical priors in the MoG model with conjugate priors. Note the dependency of the distribution of the component mean on the component precision. Variables are labelled below by the name of their distribution, and the parameters of these distributions are given above.

particular value. The graphical representation of the hierarchical model is depicted in Figure 3.10.

Note in eq. (3.57) that the precision for the mean is a multiple of the component precision itself. This dependency is probably not generally desirable, but is an unavoidable consequence of requiring conjugacy.

Conditionally Conjugate DPMoG

If we remove the undesired dependency of the mean on the precision, we no longer have conjugacy. For a more realistic model, we define the prior on $\boldsymbol{\mu}_j$ to be

$$p(\boldsymbol{\mu}_j|\boldsymbol{\xi}, R) \sim \mathcal{N}(\boldsymbol{\xi}, R^{-1}) \tag{3.60}$$

whose mean vector $\boldsymbol{\xi}$ and precision matrix R are hyperparameters common to all mixture components. Keeping the Wishart prior over the precisions as in eq. (3.58), we obtain the *conditionally* conjugate model. That is, the prior of the mean is conjugate to the likelihood conditional on S and the prior of the precision is conjugate conditional on $\boldsymbol{\mu}$. See Figure 3.11 for the graphical representation of the hierarchical model.

We put hyperpriors on the hyperparameters in both prior specifications to have a robust model. We use the hierarchical model specification of Rasmussen (2000) for the conditionally conjugate model, and a similar specification for the conjugate case. Vague priors are given to the hyperparameters, some of which depend on the observations which technically they ought not to. However, only the empirical mean $\boldsymbol{\mu}_x$ and the covariance Σ_x of the data are used in such a way that the full procedure becomes invariant to

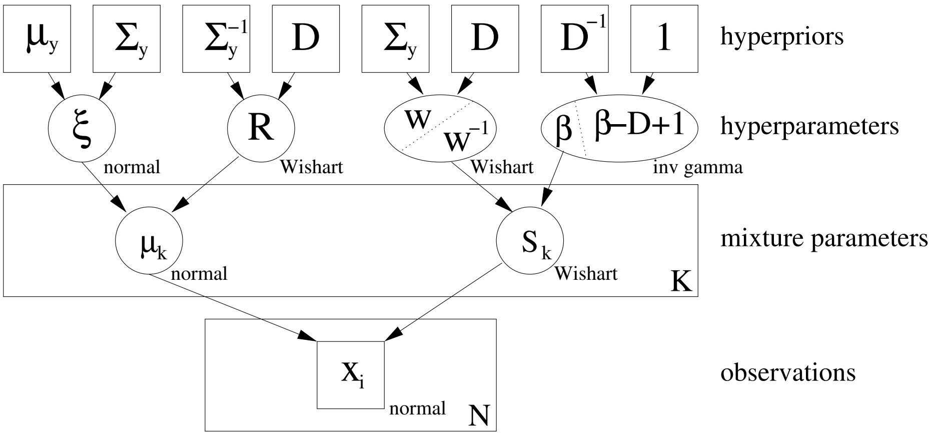

Figure 3.11: Graphical representation of the layered structure of the hierarchical priors in the conditionally conjugate model. The distribution of the component means is independent from the component precision.

translations, rotations and rescaling of the data. One could equivalently use unit priors and scale the data before the analysis since finding the overall mean and covariance of the data is not the primary concern of the analysis, rather, we wish to find structure within the data.

In detail, for both models the hyperparameters W and β associated with the component precisions S_j are given the following priors, keeping in mind that β should be greater than $D-1$:

$$W \sim \mathcal{W}(D, \tfrac{1}{D}\Sigma_x), \qquad (\tfrac{1}{\beta-D+1}) \sim \mathcal{G}(1, \tfrac{1}{D}). \tag{3.61}$$

For the conjugate model, the priors for the hyperparameters $\boldsymbol{\xi}$ and ρ associated with the mixture means $\boldsymbol{\mu}_j$ are Gaussian and gamma:

$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_x), \qquad \rho \sim \mathcal{G}(1/2,\, 1/2). \tag{3.62}$$

For the non-conjugate case, the component means have a mean vector $\boldsymbol{\xi}$ and a precision matrix R as hyperparameters. We put a Gaussian prior on $\boldsymbol{\xi}$ and a Wishart prior on the precision matrix:

$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_x), \qquad R \sim \mathcal{W}\big(D, (D\Sigma_x)^{-1}\big). \tag{3.63}$$

3.3.2 Inference Using Gibbs Sampling

In this section, we describe the MCMC algorithms we utilize for inference on the models. The Markov chain relies on Gibbs updates, where each variable is updated in turn by sampling from its posterior distribution conditional on all other variables. We repeatedly sample the parameters, hyperparameters and the indicator variables from their posterior distributions conditional on all other variables. As a general summary, we iterate:

– Update mixture parameters (mean and precision)
– Update hyperparameters
– Update the indicators, conditional on the other indicators and the (hyper)parameters
– Update the DP concentration parameter α

For the models we consider, the conditional posteriors for all parameters and hyperparameters except for α, β and the indicator variables c_i are of standard form, thus can be sampled from easily. The conditional posteriors of $\log(\alpha)$ and $\log(\beta)$ are both log-concave, so they can be updated using Adaptive Rejection Sampling (ARS) (Gilks and Wild, 1992) as suggested in (Rasmussen, 2000). We have described several algorithms for updating the indicator variables of the DPM models in Section 3.2. For the conjugate case, we use Algorithm 3 described in Section 3.2.1 that makes full use of conjugacy to integrate out the component parameters, leaving only the indicator variables as the state of the Markov chain. For the conditionally conjugate case, we use Algorithm 5 described in Section 3.2.2 utilizing only one auxiliary variable.

The likelihood for components that have observations associated with them is given by the parameters of that component, and the likelihood pertaining to currently inactive classes (which have no mixture parameters associated with them) is obtained through integration over the prior distribution. The conditional posterior class probabilities are calculated by multiplying the likelihood term by the prior.

The conditional posterior class probabilities for the DPMoG are:

$$\begin{aligned}\text{components for which } n_{-i,j} > 0\text{: } & p(c_i = j|\mathbf{c}_{-i}, \boldsymbol{\mu}_j, S_j, \alpha) \\ & \propto p(c_i = j|\mathbf{c}_{-i}, \alpha) p(\mathbf{x}_i|\boldsymbol{\mu}_j, S_j) \\ & \propto \frac{n_{-i,j}}{n-1+\alpha}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, S_j),\end{aligned} \tag{3.64a}$$

$$\begin{aligned}&\text{all others combined: } p(c_i \neq c_{i'} \text{ for all } i \neq i'|\mathbf{c}_{-i}, \boldsymbol{\xi}, \rho, \beta, W, \alpha) \\ &\propto \frac{\alpha}{n-1+\alpha} \times \int p(\mathbf{x}_i|\boldsymbol{\mu}, S) p(\boldsymbol{\mu}, S|\boldsymbol{\xi}, \rho, \beta, W) d\boldsymbol{\mu} dS.\end{aligned} \tag{3.64b}$$

We can evaluate the above integral to obtain the conditional posterior of the inactive classes in the conjugate case, but it is analytically intractable in the non-conjugate case.

Conjugate Model

When the priors are conjugate, the integral in eq. (3.64b) is analytically tractable. In fact, even for the active classes, we can marginalize out the component parameters using an integral over their posterior, by analogy with the inactive classes. Thus, in all cases the log likelihood term is:

$$\begin{aligned}\log p(\mathbf{x}_i \mid c_{-i}, \rho, \boldsymbol{\xi}, \beta, W) = &-\tfrac{D}{2}\log\pi + \tfrac{D}{2}\log\frac{\rho+n_j}{\rho+n_j+1}\\ &+\log\Gamma(\frac{\beta+n_j+1}{2}) - \log\Gamma(\frac{\beta+n_j+1-D}{2})\\ &+\tfrac{\beta+n_j}{2}\log|W^*| - \tfrac{\beta+n_j+1}{2}\log|W^* + \tfrac{\rho+n_j}{\rho+n_j+1}(\mathbf{x}_i-\boldsymbol{\xi}^*)(\mathbf{x}_i-\boldsymbol{\xi}^*)^T|,\end{aligned} \tag{3.65}$$

where

$$\boldsymbol{\xi}^* = (\rho\boldsymbol{\xi} + \sum_{l:c_l=j}\mathbf{x}_l)\Big/(\rho+n_j)$$

and

$$W^* \;=\; \beta W + \rho\boldsymbol{\xi}\boldsymbol{\xi}^T + \sum_{l:c_l=j}\mathbf{x}_l\mathbf{x}_l^T - (\rho+n_j)\boldsymbol{\xi}^*\boldsymbol{\xi}^{*T},$$

which simplifies considerably for the inactive classes.

The sampling iterations become:

– Gibbs sample $\boldsymbol{\mu}_j$ and S_j conditional on the data, the indicators and the hyperparameters
– Update hyperparameters conditional on $\boldsymbol{\mu}_j$ and S_j
– Remove the parameters, $\boldsymbol{\mu}_j$ and S_j from representation
– Gibbs sample for each indicator variable, conditional on the data, the other indicators and the hyperparameters
– Sample for the DP concentration α, using ARS

Conditionally Conjugate Model

As a consequence of not using fully conjugate priors, the posterior conditional class probabilities for inactive classes cannot be computed analytically. Here, we give details for using the auxiliary variable sampling scheme of Neal (2000), summarized in Section 3.2.2 as Algorithm 5, and also show how to improve this algorithm by making use of the conditional conjugacy.

`SampleBoth`
For each observation $\mathbf{x}_i$ in turn, the updates are performed by: “invent” auxiliary classes by picking means $\boldsymbol{\mu}_j$ and precisions S_j from their priors. We update c_i using Gibbs sampling (i.e., sample from the discrete conditional posterior class distribution), and finally remove the components that are no longer associated with any observations. This is the algorithm by (Neal, 2000). Here, to emphasize the difference between the other sampling schemes that we will describe, we call it the `SampleBoth` scheme since both means and precisions are sampled from their priors to *represent* the inactive components.

The sampling iterations are as follows:

– Gibbs sample $\boldsymbol{\mu}_j$ and S_j conditional on the indicators and hyperparameters .
– Update hyperparameters conditional on $\boldsymbol{\mu}_j$ and S_j

– For each indicator variable:

– If c_i is a singleton, assign its parameters $\boldsymbol{\mu}_{c_i}$ and S_{c_i} to one of the auxiliary parameter pairs.

– Invent other auxiliary components by sampling values for $\boldsymbol{\mu}_j$ and S_j from their respective priors.

– Update the indicator variable, conditional on the data, the other indicators and hyperparameters.

– Discard the empty components.

– Update the DP concentration parameter α.

The integral we want to evaluate is over two parameters, $\boldsymbol{\mu}_j$ and S_j. Exploiting the *conditional* conjugacy, it is possible to integrate over one of these parameters given the other. Thus, we can pick only one of the parameters randomly from its prior, and integrate over the other, which might lead to faster mixing. The log likelihood for the `SampleMu` and the `SampleS` schemes are as follows:

`SampleMu`

Sampling $\boldsymbol{\mu}_j$ from its prior and integrating over S_j gives the conditional log likelihood:

$$\begin{aligned}\log p(\mathbf{x}_i|\mathbf{c}_{-i},\boldsymbol{\mu}_j,\beta,W) \;=\; & -\tfrac{D}{2}\log\pi + \tfrac{\beta+n_j}{2}\log|W^*| - \tfrac{\beta+n_j+1}{2}\log|W^*+\mathbf{x}_i\mathbf{x}_i^T| \\ & + \log\Gamma(\frac{\beta+n_j+1}{2}) - \log\Gamma(\frac{\beta+n_j+1-D}{2}),\end{aligned} \tag{3.66}$$

where $W^* = \beta W + \sum_{l:c_l=j}(\mathbf{x}_l-\boldsymbol{\mu}_j)(\mathbf{x}_l-\boldsymbol{\mu}_j)^T$.

The sampling steps for the indicator variables are:

– Remove all precision parameters S_j from the representation.

– If c_i is a singleton, assign its mean parameter $\boldsymbol{\mu}_{c_i}$ to one of the auxiliary parameters.

– Invent other auxiliary components by sampling values for the component mean $\boldsymbol{\mu}_j$ from its prior.

– Update the indicator variable, conditional on the data, the other indicators, the component means and hyperparameters using the likelihood given in eq. (3.66).

– Discard the empty components.

`SampleS`

Sampling S_j from its prior and integrating over $\boldsymbol{\mu}_j$, the conditional log likelihood becomes:

$$\begin{aligned}\log p(\mathbf{x}_i|\mathbf{c}_{-i},S_j,R,\boldsymbol{\xi}) \;=\; & -\tfrac{D}{2}\log(2\pi) - \tfrac{1}{2}\mathbf{x}_i^T S_j\mathbf{x}_i \\ & + \tfrac{1}{2}(\boldsymbol{\xi}^*+S_j\mathbf{x}_i)^T\Big((n_j+1)S_j+R\Big)^{-1}(\boldsymbol{\xi}^*+S_j\mathbf{x}_i) \\ & - \tfrac{1}{2}\boldsymbol{\xi}^{*T}\Big(n_jS_j+R\Big)^{-1}\boldsymbol{\xi}^* + \tfrac{1}{2}\log\frac{|S_j||n_jS_j+R|}{|(n_j+1)S_j+R|},\end{aligned} \tag{3.67}$$

where $\boldsymbol{\xi}^* = S_j\sum_{l:c_l=j}\mathbf{x}_j + R\boldsymbol{\xi}$.

The sampling steps for the indicator variables are:
– Remove the component means μ_j from the representation.
– If c_i is a singleton, assign its precision S_{c_i} to one of the auxiliary parameters.
– Invent other auxiliary components by sampling values for the component precision S_j from its prior.
– Update the indicator variable, conditional on the data, the other indicators, the component precisions and hyperparameters using the likelihood given in eq. (3.67).
– Discard the empty components.

Note that `SampleBoth`, `SampleMu` and `SampleS` are three different ways of doing inference for the same model since there are no approximations involved. One would expect only the mixing time to differ among these schemes, whereas the conjugate model discussed in the previous section is a different *model* since the prior distribution is different.

3.3.3 Experiments

In this section, we present results on simulated and real data sets with different dimensions to compare the predictive accuracy and computational time complexity of the different models and sampling schemes described above. The density of each data set has been estimated by the conjugate model (CDP), the three different sampling schemes for the conditionally conjugate model (CCDP), `SampleBoth, SampleMu, and SampleS`, and by kernel density estimation[4] (KDE) using Gaussian kernels.

We use the duration of consecutive eruptions of the Old Faithful geyser (Scott, 1992) as a two dimensional example, for which we can visualize the estimated densities, see Figure 3.12. Additionally, the three dimensional Spiral data set used in Rasmussen (2000), the four dimensional "Iris" data set used in (Fisher, 1936) and the 13 dimensional "Wine" data set (Forina et al., 1986) were modeled for assessing the computational cost in higher dimensions.

Convergence was determined by examining various properties of the state of the Markov chain, and mixing time was calculated as the sum of the auto-covariance coefficients of the slowest mixing quantities from lag -1000 to 1000. In all experiments, the slowest mixing quantity was found to be the number of active components. Example auto-covariance coefficients are shown in Figure 3.13. The convergence time for the CDP model is usually shorter than the `SampleBoth` scheme of CCDP but longer than the two other schemes. For the CCDP model, the `SampleBoth` scheme is the slowest both in terms of converging and mixing. `SampleS` has comparable convergence time to the `SampleMu` scheme.

The three different sampling schemes for the conditionally conjugate model all have identical equilibrium distributions, therefore the result of the conditionally conjugate model is presented only once, instead of discriminating between different schemes when the predictive densities are considered.

[4] KDE is a classical non parametric density estimation technique which places kernels on each training data point. The kernel bandwidth is adjusted separately on each dimension to obtain a smooth density estimate, by maximizing the sum of leave-one-out log densities.

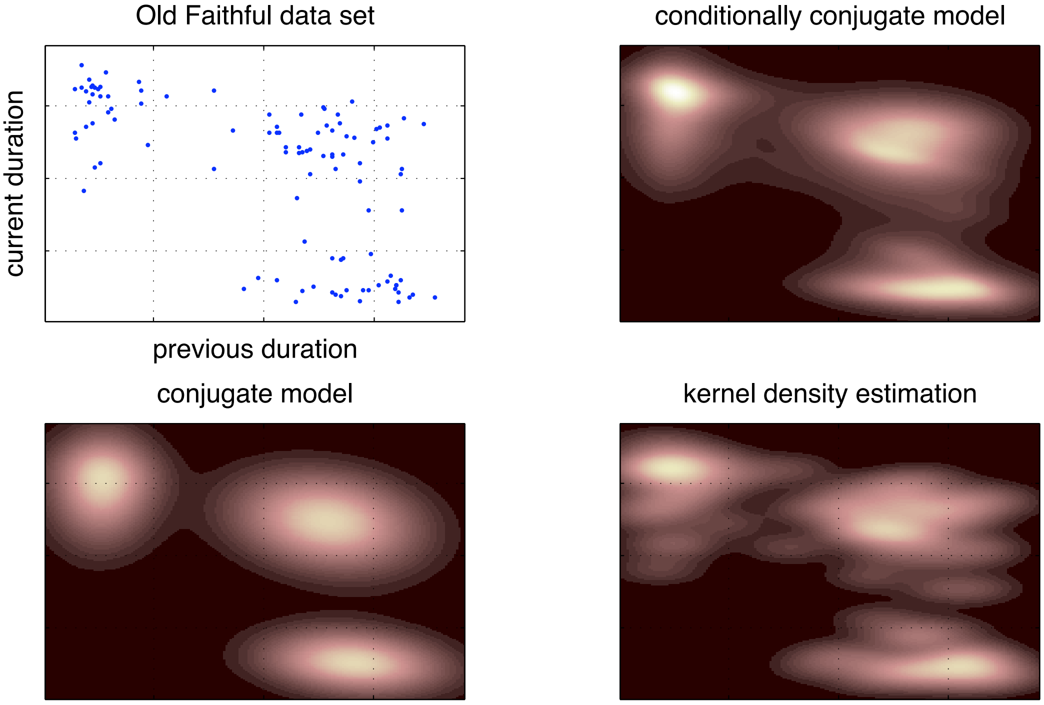

Figure 3.12: The Old Faithful geyser data set and its density modelled by CDP, CCDP and KDE. The two dimensional data consists of the durations of the consecutive eruptions of the Old Faithful geyser.

As a measure for modeling performance, we use the average leave one out predictive densities. That is, for all data sets considered, we leave out one observation, model the density using all others, and calculate the predictive density on the left-out data point. We repeat this for all data points in the training set and report the average predictive density.

The mixing of all samplers is equally fast for the two dimensional Geyser data set. There is also not a significant difference in the predictive performance, see Tables 3.1 and 3.2. However, we can see form the plots in Figure 3.12 that the resulting density estimates are different for all models.

For all data sets, the KDE model has the lowest average leave one out predictive density, and the conditionally conjugate model has the best (see Table 3.1). To compare the distribution of the leave one out densities, p-values for a paired t-test are given (Table 3.2). For the Spiral, Iris and Wine data sets, the difference between the predictive densities of KDE and both DP models were statistically significant.

The main objective of the models presented in this paper is density estimation, but the models can be used for clustering as well by observing the assignment of data points to model components. Since the number of components change over the chain, one

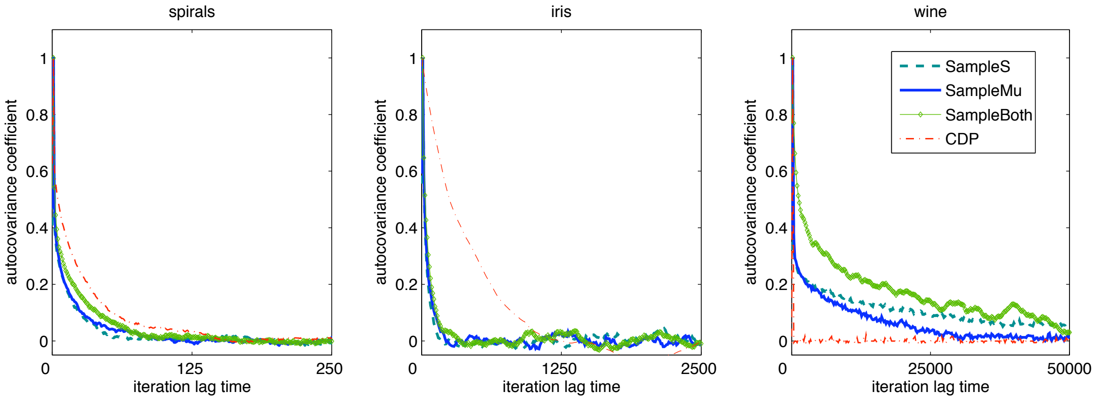

Figure 3.13: Autocorrelation coefficients of the number of active components for CDP and different sampling schemes for CCDP, for the Spiral data based on 5×10^5 iterations, the Iris data based on 10^6 iterations and Wine data based on 1.5×10^6 iterations.

would need to form a confusion matrix showing the frequency of each data pair being assigned to the same component for the entire Markov chain.

Class labels are avaliable for the Iris and Wine data sets, both data sets consisting of 3 classes. The CDP model has 3-4 active components for the Iris data and 3 active components for the Winedata. The assignment of data points to the components shows successful clustering. The CCDP model has more components on average for both data sets, but data points with different labels are not assigned to the same component, resulting in successful clustering. The Spiral data set is generated by sampling 5 points form each of the 160 Gaussians whose means lie on a spiral. For this data, the number of active components of CDP and CCDP do not go beyond 21 and 28, respectively. This is due to the assumption of independence of component means for both models, which does not hold for this data set. The data has been generated from clusters whose means lie on a spiral. The distribution of the number of active components for the

Table 3.1: Average leave one out log-predictive densities for kernel density estimation (KDE), conjugate DP mixture model (CDP), conditionally conjugate DP mixture model (CCDP) on different data sets. The ratio of the average probability of the DP model vs the KDE model is given in parenthesis. Note the increase in discrepancy as the data dimension increases.

Data set	KDE	CDP	CCDP
Geyser	-1.9058	-1.9023(1.003)	-1.8785(1.028)
Spiral	-7.2052	-7.1228(1.086)	-7.1165(1.093)
Iris	-1.8599	-1.5769(1.327)	-1.5460(1.369)
Wine	-18.9788	-17.5946(3.99)	-17.3409(5.15)

Table 3.2: Paired t-test scores of leave one out predictive densities. The test does not give enough evidence in case of the Geyser data, however it shows that KDE is statistically significantly different than both DP models for the higher dimensional data sets. Also, CDP is signifcantly different than CCDP for the Wine data.

Data set	KDE/CDP	KDE/CCDP	CDP/CCDP
Geyser	0.95	0.59	0.41
Spiral	<0.01	<0.01	0.036
Iris	<0.01	<0.01	0.099
Wine	<0.01	<0.01	<0.01

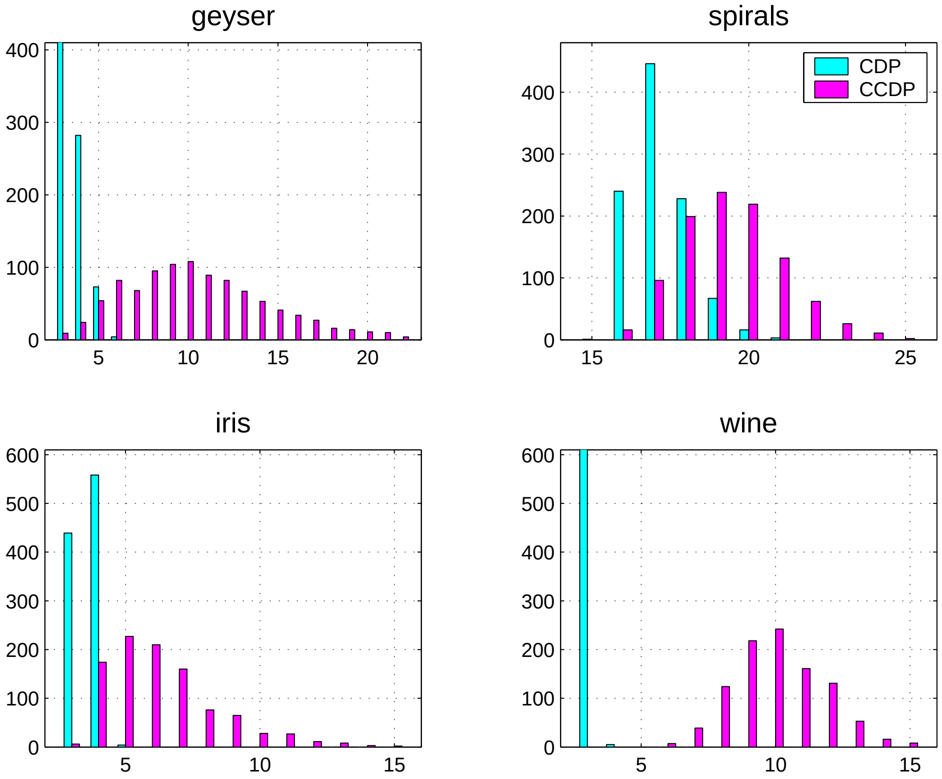

Figure 3.14: Distribution of number of active components from 1000 iterations. The CDP model favors a lower number of components for all data sets. The average number of components for the CCDP model is larger, with a more diffuse distribution. Note that histogram for the CDP model for the Geyser data set and the Winedata set has been cut off on the y-axis

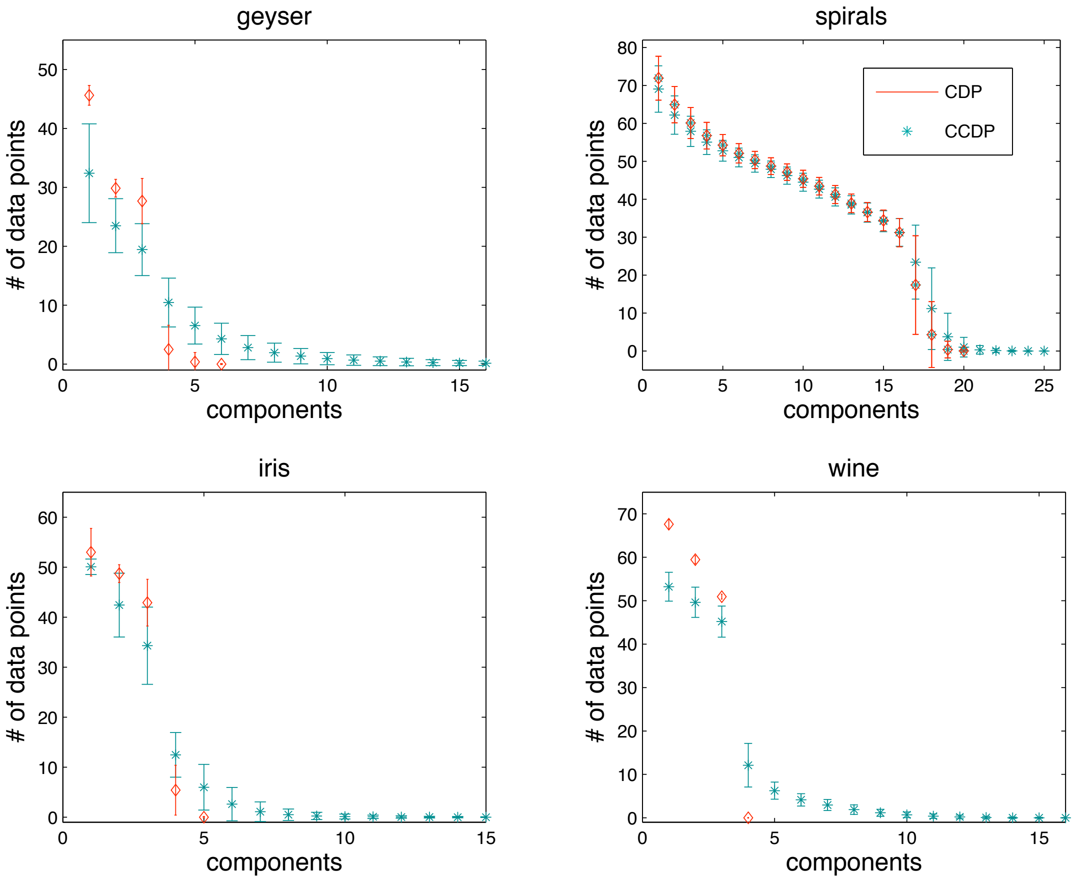

Figure 3.15: Number of data points assigned to the components averaged over different positions in the chain. The standard deviations are indicated by the error bars. Note the existence of many small components for the CCDP model

different models are depicted in Figure 3.14. This figure shows that the distribution of the number of components used by the CCDP is much broader and centered on higher values. The number of data points assigned to the components averaged over different positions in the chain is depicted in Figure 3.15.

3.3.4 Conclusions

The Dirichlet process mixtures of Gaussians model is one of the most widely used DPM models. We have presented and compared the conjugate and conditionally conjugate hierarchical Dirichlet process Gaussian mixture models. We presented two new MCMC schemes that improve the convergence and mixing properties of the MCMC algorithm of Neal (2000) for the conditionally conjugate Dirichlet process mixture model. The convergence and mixing properties of the samplers have been demonstrated on example data sets. The modeling properties of the conjugate and the conditionally conjugate model have been empirically compared. The predictive accuracy of the CCDP model is found to be better than the CDP model for all data sets considered, the difference being larger in high dimensions. It is also interesting to note that the CDP model tends to use less components than the CCDP model.

In the light of the empirical results, we conclude that marginalizing over one of the parameters by exploiting conditional conjugacy leads to considerably faster mixing in the conditionally conjugate model. When using this trick, the fully conjugate model is not necessarily computationally cheaper. The DP Gaussian mixture model with the more flexible prior specification (conditionally conjugate prior) can be used on higher dimensional density estimation problems, resulting in better density estimates than the model with conjugate prior specification.

3.4 Dirichlet Process Mixtures of Factor Analyzers

Factor analysis (FA) is a well known latent variable model that models the correlation structure in the data. The mixture of factor analyzers (MFA) model combines FA with a mixture model, allowing each component to have a different latent representation.

The generative model for FA is given by $\mathbf{x} = \Lambda\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$, where $\mathbf{z}$ is the hidden factor, Λ the factor loading matrix, and $\boldsymbol{\varepsilon}$ the measurement noise. The factors and noise are assumed to be Gaussian distributed, $\mathbf{z} \sim \mathcal{N}(0, \mathrm{I})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \Psi)$ where Ψ is a diagonal matrix. Therefore, $\mathbf{x}$ is also Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\Lambda\Lambda^T + \Psi$. Considering a mixture of factor analyzers (MFA) with K components, the data distribution becomes

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{\mu}_j, \Lambda_j {\Lambda_j}^T + \Psi), \tag{3.68}$$

where π_j denote the mixing proportions. Each component has a separate mean parameter $\boldsymbol{\mu}_j$ and a factor loading matrix Λ_j. The diagonal *uniqueness* matrix Ψ is common for all components, capturing the measurement noise in the data.

FA is generally used as a dimensionality reduction technique by assuming that the dimension of the latent factors $\mathbf{z}$ is much less than the data dimension, $Q \ll D$. For this case, a FA can be interpreted as a Gaussian with a constrained covariance matrix. Assuming that most of the structure lies in a lower dimensional space, MFA can be used to model high dimensional data as a mixture of Gaussians with less computational cost.

Analytical inference is not possible for the MFA models but maximum likelihood methods such as expectation maximization (EM) can be used to make point estimates for the parameters (Ghahramani and Hinton, 1996) or MCMC methods can be used for obtaining the posterior distribution (Utsugi and Kumagai, 2001). These algorithms are straightforward to apply for fixed latent dimension and a fixed number of components. Deciding on the latent dimension and the number of components to be used is an important modeling decision for MFA which is usually dealt with by using cross validation. Alternatively, learning the latent dimension or the number of components can be included in the inference.

Bayesian inference on the latent dimension of the FA model has been studied by Lopes and West (2004) using reversible jump MCMC (RJMCMC).

Ghahramani and Beal (2000) formulate a variational method for learning the MFA model. They form a hierarchical MFA model by placing priors on the component parameters and hyperpriors on some of the hyperparameters and do inference on the model using a variational approximation to the log marginal likelihood. The dimension of hidden factors for each component is determined by automatic relevance determination (ARD) (MacKay, 1994; Neal, 1996), and the number of components are determined by birth-death moves. The drawback of this method is that it is very sensitive to random initialization of the parameters and since it is an approximation, it does not guarantee finding the exact solution.

Fokoué and Titterington (2003) present a method for inferring both the latent dimension and the number of components in an MFA model using birth and death MCMC (BDMCMC) of Stephens (2000) which is an alternative to RJMCMC. They treat both the hidden dimension and the number of components as a parameter of the model and do inference on both using BDMCMC.

For these approaches, the number of components can be seen as a parameter of the model which is inferred from the data. The DP prior allows a nonparametric Bayesian formulation of the MFA model, eliminating the need to do inference on the number of components necessary to represent the data.

In this section, we introduce the Dirichlet process mixtures of factor analyzers model (DPMFA), a FA model with a DP prior over the distribution of the model parameters.

Dirichlet Process Mixtures of Factor Analysers

We start with the distributional assumptions of (Ghahramani and Beal, 2000) for the parametric MFA model, and form the Dirichlet process MFA model by taking $K \to \infty$. In detail, the prior for the means $\boldsymbol{\mu}_j$ is Gaussian,

$$\boldsymbol{\mu}_j \sim \mathcal{N}(\boldsymbol{\xi}, R^{-1}), \tag{3.69}$$

with common hyperparameters for all components. Inverse gamma priors are put on the entries σ_d^2 of the diagonal matrix Ψ,

$$\sigma_d^2 \sim \mathcal{IG}\left(\beta, \beta w\right). \tag{3.70}$$

On the columns of the factor loading matrix Λ of each component, we put a zero mean Gaussian with a different precision parameter ν_q,

$$\mathbf{\Lambda}_{\cdot q}^{j} \sim \mathcal{N}\left(0, I/\nu_q\right), \tag{3.71}$$

where $q = 1, \ldots, Q$ is the index for the hidden factor dimension and $\mathbf{\Lambda}_{\cdot q}^{j}$ denotes the column q of the factor loading matrix for component j. We let the hyperparameters ν_q of different latent dimensions vary while each hyperparameter is shared among the mixture components. This allows sharing information about the latent dimensionality between components. The number of factors is set to a fixed value Q for all components, but assigning each column to have a different precision, the effective latent dimension adjusts to the data by ARD. That is, a large value of ν_q would force the $\mathbf{\Lambda}_{\cdot q}^{j}$ to be small, not contributing much to the covariance.

The top-level priors for the hyperparameters $\boldsymbol{\xi}$ and R associated with the mixture means $\boldsymbol{\mu}_j$ are Gaussian and Wishart:

$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_x), \qquad R \sim \mathcal{W}\big(D, (D\Sigma_x)^{-1}\big), \tag{3.72}$$

where $\boldsymbol{\mu}_x$ and Σ_x are data mean and covariance as in the previous section. We put gamma priors on the precisions ν_q of the columns of the factor loading matrices,

$$\nu_q \sim \mathcal{G}(1/2,\, 1/2). \tag{3.73}$$

The hyperparameters w and β associated with the diagonal entries of Ψ are given gamma and inverse gamma priors respectively:

$$w \sim \mathcal{G}\left(1/2,\, \sigma_x^{-2}\right), \qquad \beta \sim \mathcal{IG}\left(1/2,\, 1/2\right). \tag{3.74}$$

The graphical representation of the hierarchical model is depicted in Figure 3.16.

We obtain the DPM model by putting a symmetric Dirichlet prior with parameters α/K on the distribution of the mixing proportions π_j, and taking the limit $K \to \infty$. An inverse-gamma prior is assumed on the DP parameter α,

$$\alpha \sim \mathcal{IG}\left(1/2,\, 1/2\right). \tag{3.75}$$

MCMC Inference

The parameters defining the covariance of the mixture components — i.e., the factor loading matrices Λ_j and the uniqueness matrix Ψ — appear together in the likelihood, eq. (3.68). To decouple these parameters and update them separately, we need to

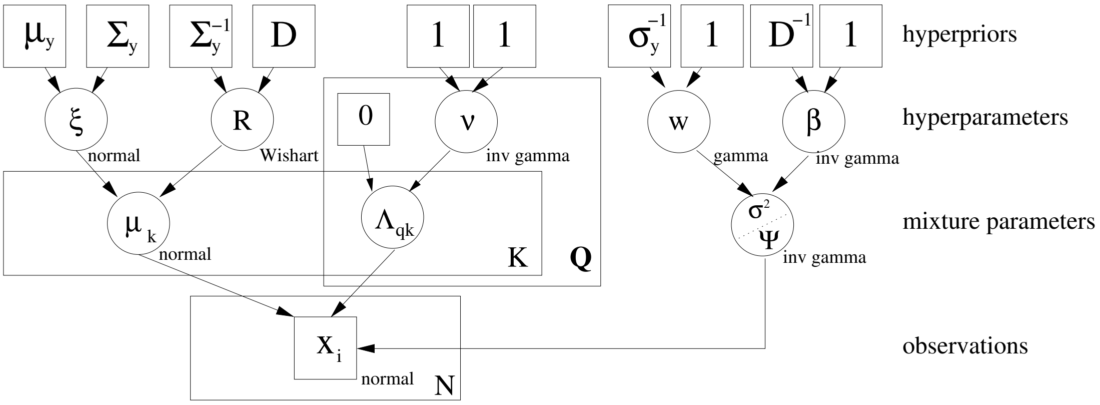

Figure 3.16: Graphical representation of the layered structure of the hierarchical priors in the mixtures of factor analyzers model. Variables are labelled below by the name of their distribution, and the parameters of these distributions are given above. The number of observations (N), number of mixture components (K) and the latent factor dimension (Q) are denoted by the numerals in the lower left hand corner of the large rectangles.

condition on the latent factor $\mathbf{z}$ which results in the likelihood:

$$p(\mathbf{x}|\mathbf{z}, c_j, \boldsymbol{\mu}_j, \Lambda_j, \Psi) = \mathcal{N}(\boldsymbol{\mu}_j + \Lambda_j \mathbf{z}, \Psi) \tag{3.76}$$

This function factorizes over the data dimensions. To make use of this factorization, we can express the prior distribution on the rows of Λ as $p(\boldsymbol{\Lambda}^j_{d\cdot}) \sim \mathcal{N}\left(0, \Upsilon_j{}^{-1}\right)$, where $\boldsymbol{\Lambda}^j_{d\cdot}$ denotes the dth row of the jth factor loading matrix and Υ_j is the diagonal matrix which has $\nu_1, \ldots, \nu_Q$ as its entries.

The likelihood function in eq. (3.76) is used to compute the posteriors of Λ_j, Ψ and $\mathbf{z}$. Once Λ_j and Ψ are updated conditioned on $\mathbf{z}$, we can compute the covariance matrix $\Sigma_j = \Lambda_j \Lambda_j{}^T + \Psi$ and condition on the full covariance to update the means and the indicators using eq. (3.68) with the hidden factors integrated out. The conditional posteriors for these variables are of standard form and therefore can be sampled from easily.

The conditional posterior of the indicator c_i is obtained by combining the prior of the components with the likelihood. The likelihood of the components that have data (other than $\mathbf{x}_i$) associated with them is Gaussian with mean $\boldsymbol{\mu}_j$ and covariance $\Lambda\Lambda_j{}^T + \Psi$. The likelihood pertaining to the remaining infinitely many components is obtained by integrating over the prior distribution of the parameters,

$$\begin{aligned} p(c_i \neq c_{i'} \text{ for all } i \neq i' | \mathbf{c}_{-i}, \boldsymbol{\xi}, R, \boldsymbol{\nu}, \beta, w, \alpha) & \\ \propto \frac{\alpha}{n-1+\alpha} \times \int p(\mathbf{x}_i|\boldsymbol{\mu}, \Lambda, \Psi) p(\boldsymbol{\mu}, \Lambda|\boldsymbol{\xi}, R, \boldsymbol{\nu}) d\boldsymbol{\mu} d\Lambda. & \end{aligned} \tag{3.77}$$

This integral is not analytically tractable, therefore we need to use an inference technique that can deal with non-conjugacy. We choose to use the auxiliary variable method of Neal (2000) which is discussed above also for inference on the DPMoG model. Using this method, the effect of the infinitely many components is represented by the auxiliary variables and the integral is avoided. We note that the mean can be integrated out, thus we would need to sample only the factor loading matrix from the prior, similar to the `SampleS` scheme for the DPMoG. We use this marginalization that leads to a considerable speedup in mixing of the chain.

In the next section, we demonstrate the modeling performance of the DPMFA model on a challenging clustering problem.

3.4.1 Spike Sorting Using DPMFA

Studying the spiking activity of neurons is important for understanding the physiological functions of the brain. Although intracellular recordings from a single neuron provide good quality signals, recording with an intracellular electrode in awake behaving animals is extremely difficult. Furthermore, recording from multiple cells at a time is desirable since it provides information about the interaction between the neurons. Extracellular electrodes introduced into the brain isolating a single neuron for each electrode have been successfully used for years for this purpose, see for example Evarts (1968). More recent work has focused on recording simultaneously from multiple neurons in order to study their interactions. Electrodes placed in the extracellular medium can record the activity of multiple nearby neurons but this leads to the question of distinguishing between the activity of individual neurons, a problem that is known as spike sorting.

Recording with multi-tip electrodes improves the identification of individual neurons compared to standard single-tip electrodes (McNaughton et al. (1983); Recce and O'Keefe (1989)). Under the assumption that the extracellular space is electrically homogeneous, four-tip electrodes (tetrodes) provide the minimal number of recording channels necessary to identify the spatial position of a source based on the relative spike amplitudes on different electrodes.

Spike sorting is usually done in three steps, namely spike detection, feature extraction and clustering. Determination of the occurrence of spikes, which is usually achieved by high-pass filtering followed by thresholding is known as the spike detection step. The spikes produced by a particular neuron have stereotypical waveforms. The difference in the features of the waveforms allows distinguishing between the activities of different neurons. In the feature extraction stage, a feature vector for each spike is calculated and clustering is done on this low dimensional feature space. Spike height, width and the peak-to-peak amplitude are some features that can be used for clustering. The clustering step involves identifying the number of sources and assigning each detected spike to one of these. In most laboratories the clustering is done manually, usually using a small number of features in order to make visualisation possible. There are also automatic spike sorting techniques that have been proposed which use different methods for feature extraction and clustering. See (Lewicki, 1998) for a review of spike sorting techniques.

A problem in spike sorting is that the true labels for the recorded data cannot be

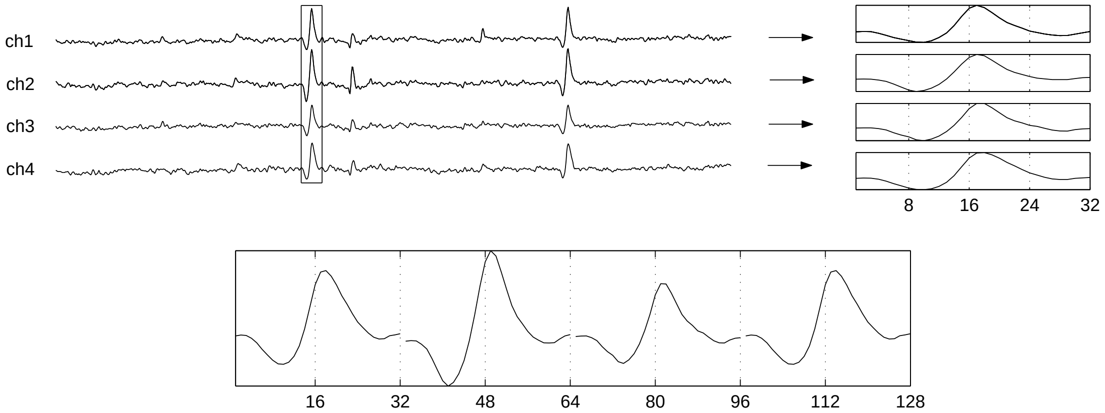

Figure 3.17: Data recording and representation: Extracellular waveforms are recorded with a tetrode. Every time the signal exceeded the threshold in one of the channels, a window of 1ms around this event was extracted from the recordings of each channel and joined to form the data vectors.

known without the verification of intracellular recordings, which makes it very hard to evaluate the results obtained by any clustering technique. Furthermore, the number of different neurons that contribute to the recorded activity is also not known. That is, we have to infer the number of clusters (different neurons) as well as the cluster assignments from the data. Therefore, we need a method that can determine the number of different neurons contributing to the data and also give satisfactory clustering results. This motivates the use of clustering models that are capable of doing automatic model selection. Nguyen et al. (2003) used reversible jump MCMC for determining the number of clusters in the spike data and Wood et al. (2006a) present spike sorting results using DPMoG. Reducing the dimensionality using feature extraction accounts to ignoring some information in the data. This improves the performance only if the ignored information is irrelevant for clustering. However, in this fully unsupervised setting, it is not easy to tell what the irrelevant information is. In general, the more relevant features we know about the signals, the better the clustering will get. Therefore, it is desirable to do the feature extraction within the clustering model. MFA can be seen as a model that combines feature extraction and clustering. We have previously used a parametric MFA model for spike sorting (Görür et al., 2004). Here, we present results of applying DPMFA to the spike sorting problem using PCA projections as well as the raw waveforms as inputs.

3.4.2 Experiments

We used data recorded with tetrodes from awake behaving macaque monkeys. The data were collected using a multi-channel data acquisition system (Cheetah Inc.). The signal was band-pass filtered between 600-6000Hz and digitised at 32kHz. The neuronal spikes were detected via an interrupt-driven, spike voltage threshold-triggered data acquisition system. That is, when a signal above threshold was detected in one of the channels, the occurrence of a spike was assumed and therefore the signal in all channels was stored within a window length of 1 ms around the triggering event with a corresponding time stamp. A sample recording from the tetrode and an extracted data vector are shown in Figure 3.17. To reduce sampling jitter, the extracted 32 dimensional signals for each channel were aligned to their peak by interpolation using cubic splines and resampling. Since the alignment was done after extracting the spike data from the recording, the dimension of the data vectors for each channel was reduced to 28 after alignment.

We did experiments on 5000 data points that had been manually clustered using peak-to-peak amplitudes of the signals in each channel. We used three different representations of the data as inputs:

1. Amplitudes: The peak-to-peak signal amplitudes from each channel (4 dimensional input vector),

2. PCA projections: The first three principal components of the waveforms from each channel (12 dimensional input vector), and

3. Waveforms: The 28 dimensional signals from each channel joined together (112 dimensional input vector).

We did clustering on the different representations using the DPMFA model and also the conjugate and the conditionally conjugate DPMoG models for comparison. We used 3, 8 and 75 latent dimensions for the DPMFA model for the 4, 12 and 112 dimensional input representations, respectively.

The conjugate DPMoG, the conditionally conjugate DPMoG and the DPMFA model on the amplitude data all give similar results. There are one small and five big clusters found by manual clustering, corresponding to activities of six different neurons. One small cluster and six big clusters have been discovered by all models. Roughly speaking, the clustering results of all models agree with the manual clustering except separating one of the big clusters into two. The confusion matrices for cluster assignments of manual clustering and the different models are depicted in Figure 3.18.

For the 12 dimensional PCA projections as inputs, the clustering results of all models were similar. Therefore for this representation, we show only the confusion matrix for the DPMFA model in Figure 3.18. The mixture models found were considerably different. Especially, there is a big difference in the average number of components employed by the models. The conjugate DPMoG model has 40 active components in average, whereas the conditionally conjugate DPMoG has 160. The conditionally conjugate model using more components is an anticipated result given the experimental results of the previous section. The average number of components used by the DPMFA model is 80.

Figure 3.19 shows the change of the number of active components over the iterations for the three different models using the PCA projections. It is interesting to see that the conjugate model converges to the stationary distribution fairly fast. The convergence and mixing for the conditionally conjugate DPMoG is slower even when using the improved sampling schemes, `SampleS` or `SampleMu`. An observation motivating the use

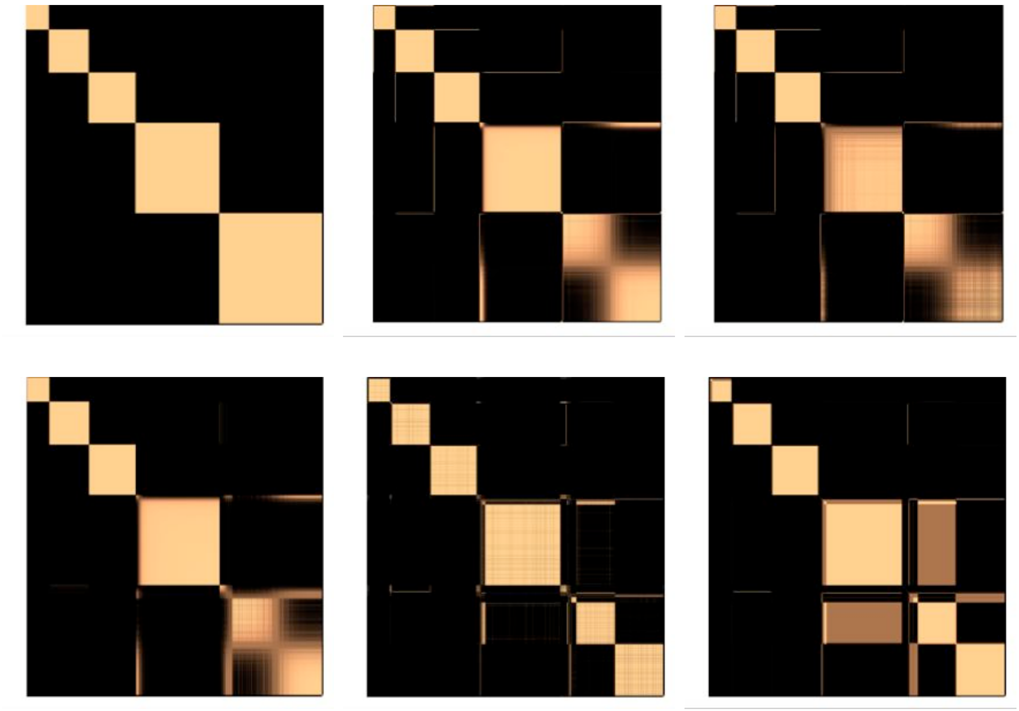

Figure 3.18: Confusion matrices for the cluster assignments. **Top row:** Using the peak-to-peak amplitudes for: the manual clustering (left), the conjugate DPMoG model (middle) and the conditionally conjugate DPMoG model (right). **Bottom row:** DPMFA results using: the peak-to-peak amplitudes (left), the 12 PCA components (middle) and the whole waveforms (right). The data is divided into one small (with 18 data points) and five big clusters (ranging from 370 to 1725 data points) by manual clustering. Note that the small cluster on the lower right hand corner of the confusion matrices is not really visible in this plot. All models find similar clustering using the amplitude data except separating one of the large clusters into two. The data is divided into more clusters by the DPMFA model using the PCA projections and the waveforms as input.

of DPMFA in higher dimensions is that the mixing for the DPFMA model is relatively fast when exploiting the conditional conjugacy.

The models were initialized with all data points assigned to the same component. We see in Figure 3.20 that the Markov chain for DPMFA using PCA components as inputs has an interesting behavior. Observing the change in the number of active components over the iterations, we see that the model initially employs around 400 components to represent the data, and starts reducing this number after some iterations, stabilizing after 2000 iterations at about 80 active components.

We also used the whole waveforms as input without a preliminary feature extraction step. Initially we tried learning all the parameters and hyperparameters, starting with a single component, like we did with the lower dimensional representations of the data.

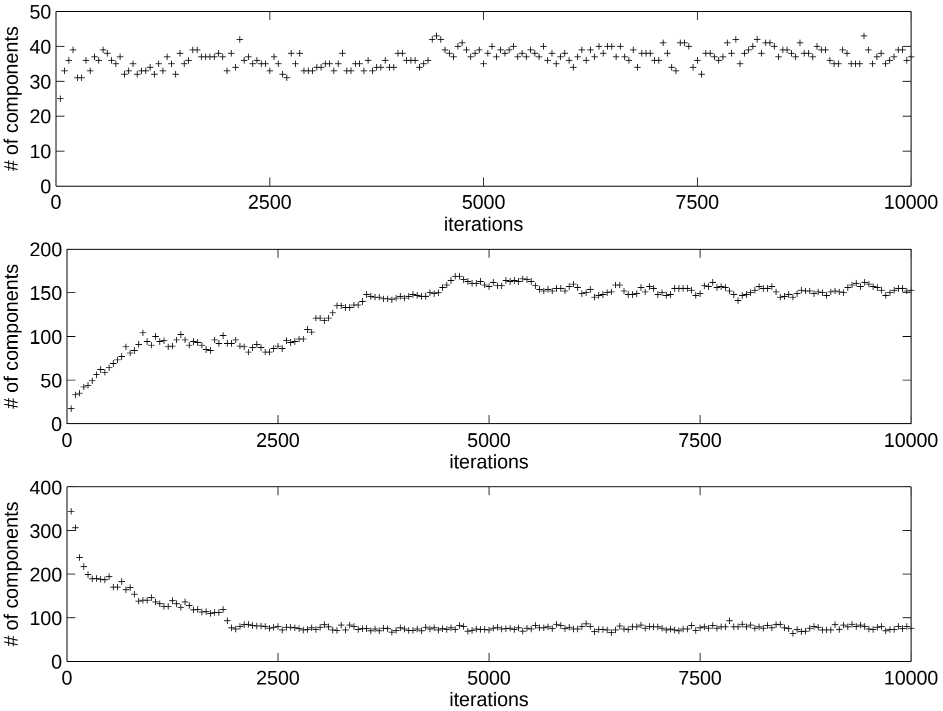

Figure 3.19: Plots showing the change in the number of clusters over the iterations using the 12 PCA components for the conjugate DPMoG (top), the conditionally conjugate DPMoG using the `SampleMu` scheme (middle) and the DPMFA model with 8 latent factors (bottom). The models are initialized with all data points assigned to a single component, and all parameters and hyperparameters are updated.

However, the sampling was not efficient for any of the models to converge. Both the conjugate and the conditionally conjugate DPMoG models stayed in the initial one component state. On the other hand, the DPMFA model employed too many active components, without converging to the stationary distribution.

A common technique when dealing with complex data using hierarchical models is to fix the hyperparameters, updating only the parameters in the beginning and start updating all unknown variables after some iterations. This stops the hyperparameters from taking values in a low probability region in an early stage of the chain and prevents the sampler from getting stuck in this low probability region. Following this approach, we tried initializing the component assignments with the manual clustering labels and not updating the hyperparameters and the indicator variables for the first 100 iterations, only updating the parameters. The conjugate and the conditionally conjugate DPMoG models could not mix at all, staying in the initialized state also in this case. The DPMFA model again employed too many components as soon as we started updating the indicator variables.

Since the concentration parameter α controls the number of active components in a DPM model, we decided to fix this parameter instead of learning it to limit the prior

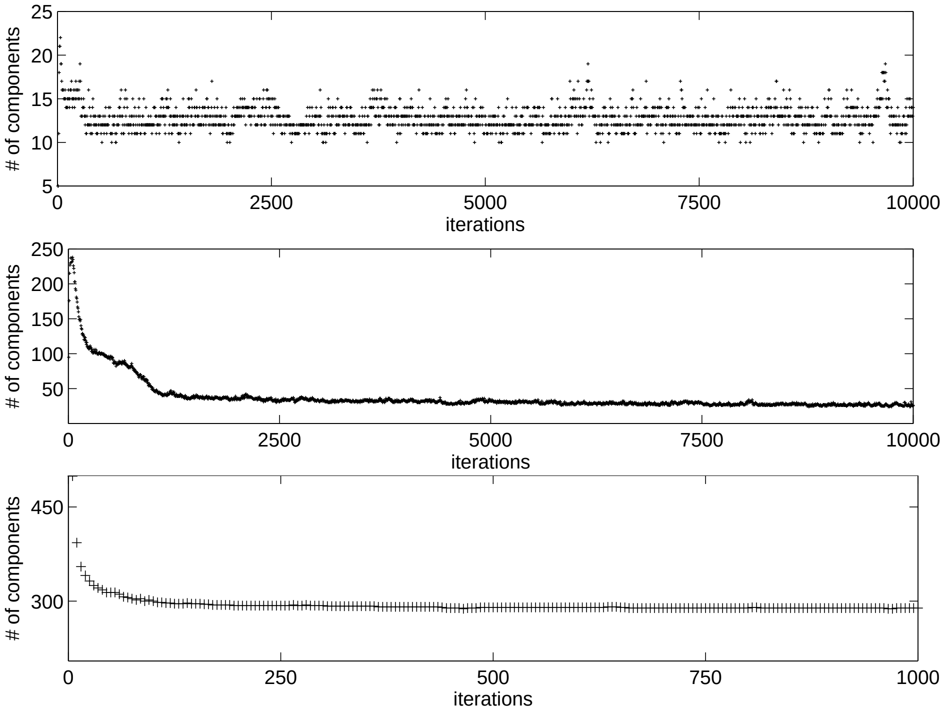

Figure 3.20: Change in the number of active components over the iterations for the DPMFA model using the peak-to-peak amplitudes (top), the 12 PCA components (middle) and the whole waveforms (bottom) with the parameter α fixed to 1.

probability of proposing new components. To check the effect of this, we also did runs with fixed α using the PCA projections. Recall that for the run on the 12 dimensions where α was updated, the number of active components first increased to 400, converging to around 80 components after 2000 iterations, see Figure 3.19. Interestingly, we observe the similar behavior even when α is fixed to 1, see Figure 3.20. For this case, the number of components increases to 250, and converges to around 40 after 1000 iterations for the 12 dimensional representation. For the 112 dimensional representation, the number of components initially goes up to 500, and settles around 300 after 100 iterations.

Although the number of active components is as high as 300, there are generally only 10 components with more than 5 data points. Figure 3.18 shows that the clustering generally agrees with the results of the other experiments. We show the manual clustering results in Figure 3.21 and the DPMFA results using the whole waveforms as inputs in Figure 3.22 for comparison. Note that the DPMFA finds a similar clustering for most of the data, except separating one of the big clusters into three.

Superimposed spike waveforms from each of the big clusters found by the DPMFA model are depicted in Figure 3.23. The waveforms from the three clusters c1, c2 and c3 were assumed to belong to one big cluster in manual clustering since they have similar amplitude characteristics. Attending to the shape of the waveforms reveals that they

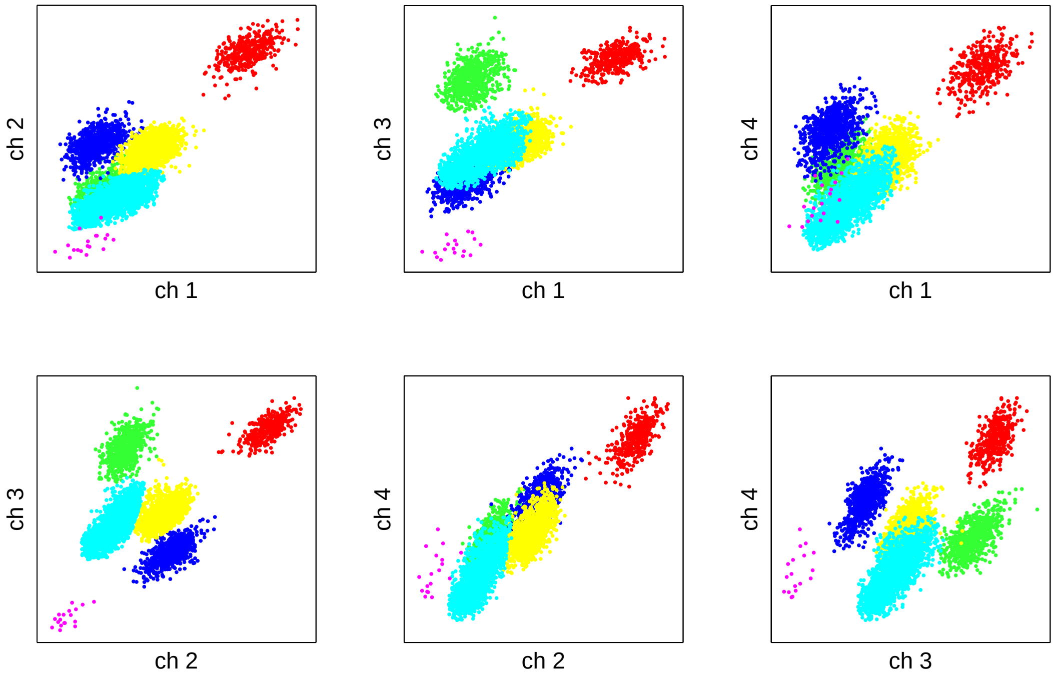

Figure 3.21: Manual spike sorting results. The two-dimensional projections of the peak-to-peak amplitudes of spikes are plotted for all combinations of 4 channels. The expert works on this representation, manually drawing boundaries around regions that may correspond to the activity of a single neuron. After cluster assignments, refractoriness is checked to verify the clustering. Each color represents a different cluster.

should belong to different clusters. Note that the waveforms assigned to the clusters c6 and c7 appear to be very similar, thus we may assume that due to the incremental updates, the sampler fails to merge these two components together which in fact belong to the same cluster (this is the case also for the the waveforms assigned to c8 and c9).

3.4.3 Conclusions

We presented experimental results for spike sorting using the conjugate DPMoG, the conditionally conjugate DPMoG and the DPMFA models on different data representations. Although the modeled density differs, the clustering results of all three models are similar on the amplitude data and on the PCA projections. In all models, the mixing of the Markov chain was poor when the whole waveforms were used as inputs. This result can be attributed to the high dimensionality of the data. Although none of the models seem to be mixing, the DPMFA model could move enough to explore a mode of the posterior whereas the DPMoG could not move at all from the initial point. This shows that the DPMFA can handle higher dimensional data more easily. However, the mixing is not good enough for practical use of the model due to the incremental updates using

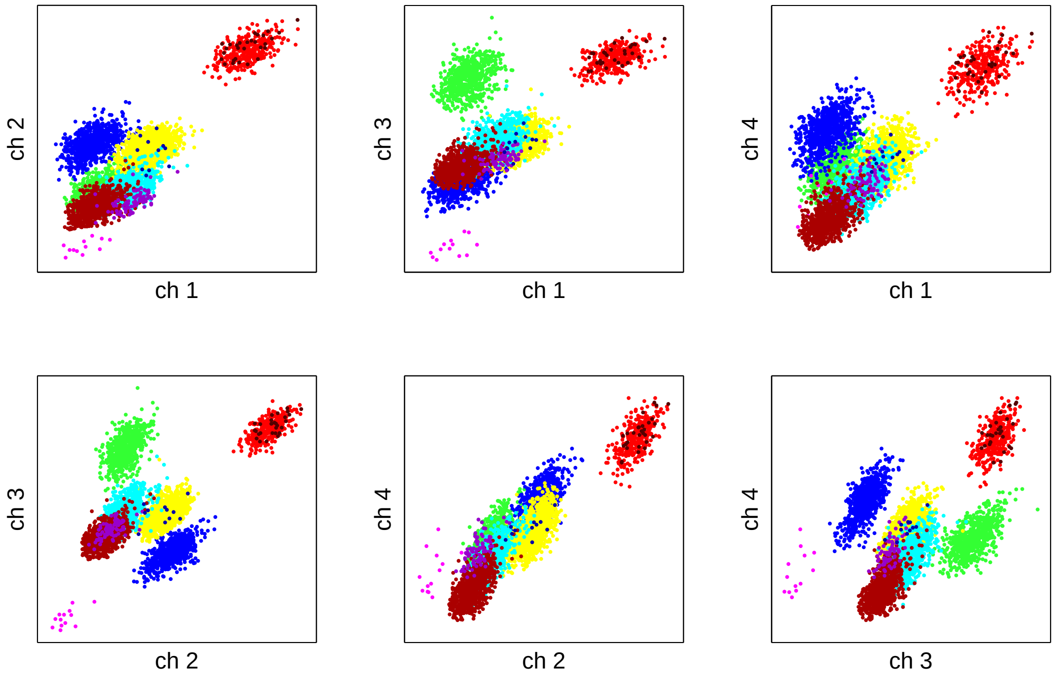

Figure 3.22: Peak-to-peak amplitude plots showing the clustering results of DPMFA using the whole waveforms. Note that one big cluster of manual clustering (light blue) is divided into three clusters with this model (light blue, brown and purple).

Gibbs sampling.

The behavior of the Markov chain for the DPMFA model on the PCA components and the whole waveforms is interesting. No matter if initialized with a single component or many components, the model first explores the space by introducing many components and eventually reduces the number of active components to give a good representation of the data. The sampling can handle this for the lower dimensional PCA inputs, however for the 112 dimensional inputs, it fails to converge to a stationary distribution when all variables are updated. For this reason, for using the waveforms as inputs, we fixed the concentration parameter α to 1. An alternative can be to have better initial values for the hyperparameters but this would involve a thorough analysis of the data.

The motivation for developing this model is to be able to apply the DPM model to high dimensional data. Experimental results show that the sampling algorithms used are not efficient enough. Using the split-merge method of Jain and Neal (2005) would speedup mixing. Furthermore, the factor loading matrix is rotation independent, therefore not unique. Restricting it to be unique, e.g. forcing it to be lower diagonal as suggested by Fokoué and Titterington (2003) may also help speedup mixing.

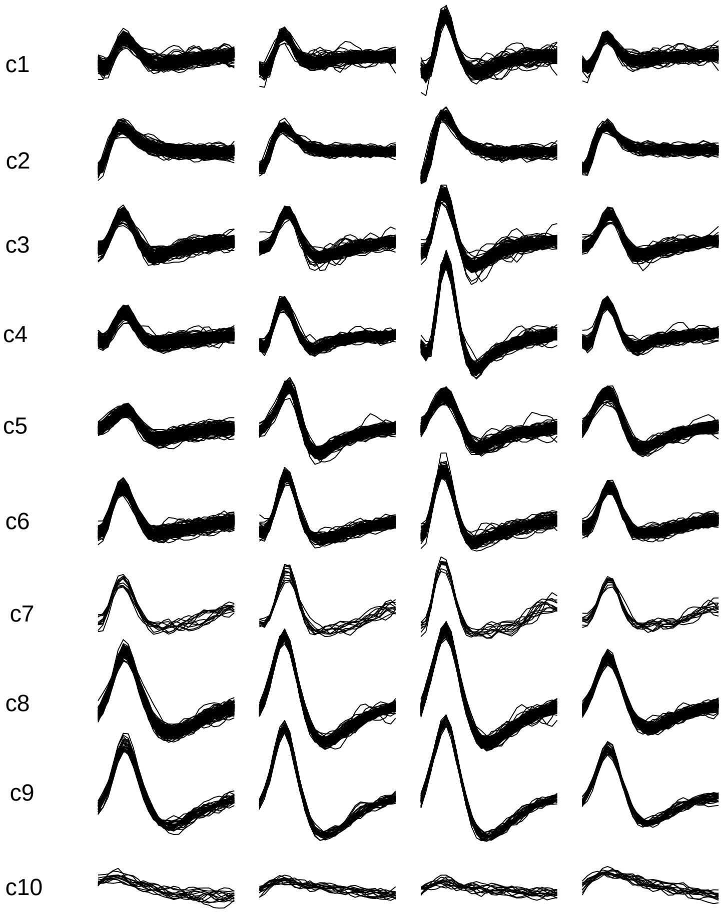

Figure 3.23: Example spike waveforms assigned to different clusters by the DPMFA model using the whole waveforms as inputs. The waveforms assigned to clusters 1, 2 and 3 have similar amplitude characteristics. Therefore, they were assumed to belong to one big cluster in manual clustering since they have similar amplitude characteristics. DPMFA can discover that they should belong to different clusters by using the waveform information. On the other hand, the waveforms assigned to the clusters c6 and c7 appear to be very similar. It is possible that due to the incremental updates, the sampler fails to merge these two components together which belong to the same cluster. This is also the case for the waveforms assigned to components c8 and c9.

3.5 Discussion

In this chapter, we have considered the DPM models for density estimation and clustering. The DP has been introduced by Ferguson (1973) and has been extensively used especially since the development of MCMC methods for inference. There are several different approaches to define the DP. In the beginning of this chapter, we have summarized some of these approaches to give an insight about the DP and its properties. The different ways of defining the same distribution has lead to several different inference algorithms for DPM models. We have outlined some of the MCMC algorithms developed for inference on the DPM models. The list of algorithms that we described is not exhaustive, but we believe that they give a good overview of the development of the techniques.

We compared the DPMoG models with conjugate and conditionally conjugate base distributions. We showed that for the inference algorithm we used, mixing time of the samplers can be vastly improved for the conditionally conjugate model by integrating out one of the parameters while conditioning on the other. Using this improved sampling scheme, inference for the conditionally conjugate model is not always computationally more expensive than the conjugate one. The relative mixing performance depends on the data modeled. The modeling performance of the conditionally conjugate model for density estimation is found to be always better than the fully conjugate one, the difference being significant in higher dimensions. These results suggest that one does not have to resort to using the conjugate base distribution when it does not represent the prior beliefs.

We have introduced the DPMFA model for modeling high dimensional data that is believed to have a low dimensional representation as mixture of Gaussians with constrained covariance matrices. We have demonstrated the modeling performance of the DPMFA on a challenging clustering problem. Although the resulting clustering is successful, the incremental sampling algorithm used is not feasible for practical use of the model on high dimensional data and should be improved.

Difference between the DPM models and mixtures with finite but unknown number of components is that the DPM takes into account the possibility of a yet not observed data point to be coming from an unrepresented component through the concentration parameter of the DPM both during training and when the predictive probabilities are calculated. Although the RJMCMC or BDMCMC give a distribution over the possible number of components, once an iteration is complete and a number of components is chosen, it is only those components that are used to explain the new data. The comparison of the performance if these models remains to be an interesting question.

4 Indian Buffet Process Models

Latent, or hidden, variable models are powerful tools to model the underlying structure in data. The idea is to augment the observed data with hidden variables to obtain a powerful model with an easy to handle inference scheme. Depending on the model used, the latent variables may be interpretable or not. Unsupervised learning aims to discover the systematic component in the data to make meaningful summaries of it, such as density estimation, feature extraction or clustering.

A prominent example of discrete latent variable models are mixture models, which are also referred to as latent class models. The data is assumed to come from a mixture of distributions (generally of simple parametric form), and the latent class variables indicate which mixture component each data point is generated from. Mixture models have been widely used for density estimation and clustering. Since the classes are mutually exclusive, mixture models can be a good representation of the data if the data can be divided into homogeneous disjoint subsets.

Dirichlet process mixture (DPM) models presented in the previous chapter constitute an example of infinite latent class models, in which each data point is assumed to belong to exactly one of the classes. In this chapter, we will consider unsupervised learning using nonparametric discrete latent feature models with nonexclusive features.

A powerful extension of the latent class models is obtained by relaxing the condition of the classes being mutually exclusive, and assuming each data point to be generated by a combination of features encoded by the binary feature presence vectors. Recently, the Indian buffet process (IBP) (Griffiths and Ghahramani, 2005), which specifies a distribution over sparse binary matrices with infinitely many columns, has been defined. Inspired by the Chinese restaurant process (CRP) (Section 3.1.2), IBP is a sequential process for generating sparse binary matrices with an unbounded number of columns. In the CRP, customers decide which single table to sit at, whereas in the IBP, they decide which dishes to taste. Each customer can choose any number of offered dishes, which can be interpreted as the data points belonging to more than one class, or the data points being described with more than one feature. Thus, infinite binary latent feature (IBLF) models can be defined using the IBP as a nonparametric distribution over the matrix of features. In this chapter, we will focus on the IBP and related distributions for defining IBLF models.

The correspondences between the CRP and the IBP are not restricted to their culinary metaphorical similarities. The distribution on the binary matrices with infinitely many columns induced by the IBP can be defined in several different ways. Each of these approaches helps to show interesting properties of the distribution and reveals its connections to the Dirichlet process. As well as defining the IBP, Griffiths and Ghahramani (2005) derive the distribution defined by the IBP starting with a finite binary matrix and taking the limit as the number of columns approaches infinity. Teh, Görür, and Ghahramani (2007) construct a stick-breaking representation for the IBP, and show interesting connections between the stick lengths of the stick-breaking construction for the DP and the IBP. Thibaux and Jordan (2007) show that the underlying de Finetti mixing distribution for the IBP is the beta process (Hjort, 1990). In the next section, we present the description of the IBP and summarize these different approaches for defining the distribution.

Analytical inference is not possible in the IBLF models. Due to the similarity of the IBP and the DP, the methods for DPM models can be adjusted for the IBLF models. Similar to the DPM case, Gibbs sampling in conjugate IBLF models is fairly straightforward to implement but requiring conjugacy does not hold for many interesting IBLF models. In Section 4.2, we summarize the Gibbs sampling algorithm for conjugate IBLF models, as well as other MCMC algorithms that can be used for inference on the non-conjugate IBLF models. In Section 4.3, we apply the described algorithms for inference on a simple model on several synthetic data sets for empirically comparing the performance of the different samplers.

In Section 4.4, we use an IBLF model suggested in (Griffiths and Ghahramani, 2005) to learn the features of handwritten digits. We employ two-part latent vectors for representing the hidden features of the observations: a binary vector to indicate the presence or absence of a latent dimension and a real valued vector as the contribution of each latent dimension. Using an IBP prior over the binary part, we have an over-complete dictionary of basis functions. In fact, we have infinitely many basis functions, only a small number of which are active for each data point. We show that the model can successfully reconstruct the images by finding features that make up the digits.

Another interesting application of IBLF is modeling the choice behavior using the latent features to represent the alternatives in a choice set, Görür et al. (2006). In Section 4.5 we describe the non-parametric choice model using IBP as a prior. We show that the features that lead to the choice data can be inferred using the model.

There have been other interesting applications of IBP to define flexible latent feature models. These include protein-protein interactions (Chu et al., 2006), the structure of causal graphs (Wood et al., 2006b), dyadic data for collaborative filtering (Meeds et al., 2007), human similarity judgements (Navarro and Griffiths, 2007) and document classification (Thibaux and Jordan, 2007).

4.1 The Indian Buffet Process

The Indian buffet process (IBP) (Griffiths and Ghahramani, 2005) defines a distribution on sparse binary matrices with infinitely many columns, and one row for each observation. It is a sequential process that has been inspired by the Chinese restaurant process (see Section 3.1.2) and is described by imagining an Indian buffet restaurant offering an unbounded number of dishes. The metaphor is describing the N rows of a binary matrix Z, each row represents a customer arriving at the restaurant and the infinitely many columns of the matrix represent the dishes offered, a finite number of which will

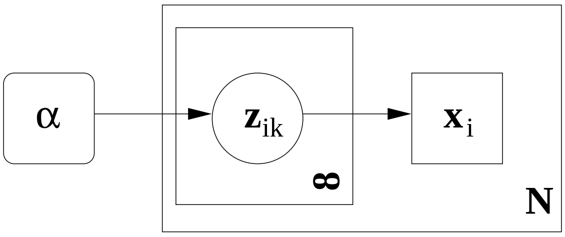

Figure 4.1: Graphical representation for the infinite binary feature model using the Indian buffet process as a prior over the features. The observations $\mathbf{x}_i$ defined by binary features (and possibly other parameters). The binary features z_{ik} are generated by an IBP(α).

be chosen by the customers. An entry $z_{ik} = 1$ means that the ith customer chose the kth dish. In a latent feature model, the customers correspond to the data points and the dishes correspond to the features describing the data. Thus, an entry of the binary matrix indicates the presence or absence of a particular feature for a data point. And the different dishes represent the (possible) parameters associated with each column of the matrix. The process has one parameter α which is a positive scalar.

Customers sequentially enter the restaurant which offers a buffet with infinitely many dishes that are arranged in a line. The first customer entering the restaurant chooses the first Poisson(α) number of dishes from the buffet. The next customer considers taking each dish sampled by the first customer with probability 1/2, then samples Poisson($\alpha/2$) number of new dishes. The ith customer goes along the part of the buffet already visited by the previous customers. He chooses dishes with probability proportional to their popularity, i.e. with probability $(\frac{m_{.<i,k}}{i})$ where

$$m_{.<i,k} = \sum_{j=1}^{i-1} z_{jk}$$

is the number of customers that sampled the kth dish before the ith customer. After going through all the previously sampled dishes, he tries a Poisson(α/i) number of new dishes. Combining the Poisson and the Bernoulli terms, the probability of a binary matrix Z generated by this process is given as

$$P(Z) = \frac{\alpha^{K^\ddagger}}{\prod_{i=1}^{N} K_*^{(i)}!} \exp\big(-\alpha H_N\big) \prod_{k=1}^{K^\ddagger} \frac{(N - m_{.,k})!(m_{.,k} - 1)!}{N!}, \tag{4.1}$$

where $K^\ddagger$ is the total number of dishes sampled, $K_*^{(i)}$ is the number of new dishes sampled by the ith customer, $m_{.,k}$ is the number of customers that has the kth dish, and $H_N = \sum_{i=1}^{N} 1/i$. The graphical representation of the generative process is depicted in Figure 4.1. Note that the dependency of $\mathbf{z}_i$ on $\mathbf{z}_{.<i}$ is not shown because it is inherent in the IBP definition.

In the CRP, exchangeability is achieved by ignoring the labels of the tables and focusing on the resulting partitioning. A similar approach can be taken for the matrices by ignoring the column order. The analogue of partitions for class assignment vectors are the equivalence classes for binary matrices. To establish a well defined distribution in the infinite limit, Griffiths and Ghahramani (2005) define equivalence classes for binary matrices with respect to a function called *left-ordered form* ($lof(\cdot)$) and focus attention on the distribution over the equivalence class of Z.

Defining the term *history of a column* of Z to refer to the binary number corresponding to that column with the first row as the most significant bit, $lof(Z)$ sorts the columns of Z from left to right according to the histories. Any two binary matrices are *lof*-equivalent if they map to the same left-ordered form. The *lof*-equivalence class $[Z]$ of a binary matrix Z is the set of binary matrices that are *lof*-equivalent to Z.

The columns of a matrix with N rows have $(2^N - 1)$ possible different histories (other than zero, which corresponds to the column of zeros). Using the decimal equivalent of the binary histories, K_h denotes the number of columns with history h. The total number of columns of Z generated by the IBP can be expressed as $K^{\ddagger} = \sum_{h=1}^{2^N-1} K_h$. Considering the *lof*-equivalence classes of the matrices, there are

$$\frac{\prod_{i=1}^{N} K_*^{(i)}!}{\prod_{h=1}^{2^N-1} K_h!}$$

matrices Z that can be generated by the sequential process that map to the equivalent matrix $[Z]$. Therefore, the distribution over the equivalent feature matrices generated by IBP becomes

$$P([Z]) = \frac{\alpha^{K^{\ddagger}}}{\prod_{h=1}^{2^N-1} K_h!} \exp\big(- \alpha H_N \big) \prod_{k=1}^{K^{\ddagger}} \frac{(N - m_{\cdot,k})!(m_{\cdot,k} - 1)!}{N!}, \tag{4.2}$$

which accounts to both the customers and the dishes being exchangeable.

Griffiths and Ghahramani (2005) also describe the "*exchangeable* Indian buffet process" that directly produces matrices of the left-ordered form, which is established by the customers attending to the histories of the dishes.

The process described has been inspired by the Chinese restaurant process. In the CRP, customers choose which table to sit at, and favoring to be social, they tend to sit at the more crowded tables. Since each customer can sit at only one table, the CRP results in a partitioning on the customers. In the IBP, customers decide which dishes to sample. They taste the more popular dishes with higher probability, and may also taste some dishes which nobody tasted before. Every customer is free to sample any number from the infinitely many dishes. Since customers can sample more than one dish, their choices does not result in a partitioning, but can be seen as binary features that are shared between data points.

Note that, in the CRP, although there are infinitely many tables available, the total number of occupied tables is limited by the number of customers. There is no such limit in IBP since customers can choose many dishes and the resulting matrix Z potentially has infinitely many non-zero columns. Nevertheless, the expected number of total dishes sampled remains finite, since the mean number of new dishes a customer samples decreases reciprocally. It was assumed that the ith customer chooses Poisson(α/i) number of new dishes. We can use the additive property of the Poisson distribution to deduce that the total number of dishes sampled (the number of non-zero columns of Z) follows a Poisson(αH_N) distribution. This means, the effective dimension of the binary matrix is determined by the IBP parameter α and the number of customers.

The exchangeability of both the customers and the dishes has been established by focusing on the equivalence classes of the generated matrices using the *lof* function. The process starts with the first customer selecting Poisson(α) number of dishes. Making use of exchangeability, any customer can be the first one to choose. With this argument, we can see that the total number of dishes sampled by each customer follows a Poisson(α) distribution. Therefore, for a matrix with N rows (N customers), the number of non-zero entries in Z follows a Poisson(αN) distribution, which implies that the IBP produces sparse matrices.

4.1.1 A Distribution on Infinite Binary Matrices

The probability distribution over $[Z]$ defined by the IBP that is given in eq. (4.2) can be derived as the limit of a distribution over finite size binary matrices when the number of columns approach infinity (Griffiths and Ghahramani, 2005). This approach is similar to deriving the equivalent distribution to the Dirichlet process mixtures by considering a mixture model with infinitely many components (see Section 3.1.5). We start with defining the distribution over a finite binary matrix Z with N rows (customers) and K columns (dishes), then take the limit $K \to \infty$. Recall that an entry $z_{ik} = 1$ means that the ith customer has sampled the kth dish, or equivalently, the ith object has the kth feature. The graphical representation of the hierarchical model is depicted in Figure 4.2.

Each entry on the kth column of the finite-dimensional matrix is assumed to have a probability μ_k of being 1,

$$(z_{ik} \,|\, \mu_k) \sim \text{Bernoulli}(\mu_k). \tag{4.3}$$

We will refer to μ_k as the *feature presence probability* for column k. Putting an independent beta prior on each μ_k,

$$\mu_k \sim \text{Beta}(\frac{\alpha}{K}, 1), \tag{4.4}$$

the posterior distribution for μ_k given the previous $i-1$ entries of the kth column is also beta by conjugacy;

$$(\mu_k \,|\, z_{jk},\ 1 \le j < i) \sim \text{Beta}(\frac{\alpha}{K} + m_{\cdot<i,k},\ i - m_{\cdot<i,k}) \tag{4.5}$$

where $m_{\cdot<i,k} = \sum_{j=1}^{i-1} z_{jk}$ as defined in the previous section.

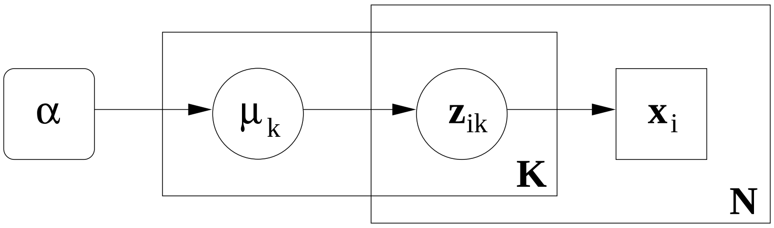

Figure 4.2: Graphical representation for the binary feature model using finite binary matrix. The parameters μ_k determine the presence or absence of the features that define the inputs.

To see the connection between the distribution over the binary matrix and the sequential Indian buffet process, we can think of starting with a matrix of zeros and sequentially setting each entry of the matrix. The incremental conditional probabilities are given as:

$$\begin{aligned} P(z_{ik} \,|\, z_{jk},\ 1 \le j < i) = &\int P(z_{ik} \,|\, \mu_k) P(\mu_k \,|\, z_{jk},\ 1 \le j < i) \mathrm{d}\mu_k \\ =&\frac{\Gamma(\alpha/K + i)}{\Gamma(m_{.<i,k} + \alpha/K)\Gamma(i - m_{.<i,k})} \int_0^1 \mu_k \, \mu_k^{m_{.<i,k}+\alpha/K-1} (1-\mu_k)^{i-m_{.<i,k}-1} \mathrm{d}\mu_k \\ =&\frac{m_{.<i,k} + \alpha/K}{i + \alpha/K}. \end{aligned} \tag{4.6}$$

Taking $K \to \infty$, the conditional probability of z_{ik} becomes

$$P(z_{ik} \,|\, z_{jk},\ 1 \le j < i) = \frac{m_{.<i,k}}{i}, \quad \text{for } m_{.<i,k} > 0 \tag{4.7}$$

since $\alpha/K \to 0$. Thus, choosing feature k for object i is proportional to the number of objects that already possess that feature. For the features that no object sampled, i.e. $m_{.<i,k} = 0$, the probability is

$$P(z_{ik} \,|\, z_{jk},\ 1 \le j < i) = \frac{\alpha/K}{i + \alpha/K}, \quad \text{for } m_{.<i,k} = 0 \tag{4.8}$$

which approaches zero as $K \to \infty$. However, note that there are infinitely many possible features with this diminishing Bernoulli probability. Considering sampling infinitely many new features, we obtain a Poisson distribution over the number of new features;

$$P(K_*^{(i)} \,|\, \alpha) = \frac{(\alpha/i)^{K_*^{(i)}} \exp(-\alpha/i)}{K_*^{(i)}!}. \tag{4.9}$$

Refer to appendix B.4 for the details of obtaining the Poisson distribution as the limit of infinitely many Bernoulli trials.

The above derivation shows how the generative process can be obtained. Griffiths and Ghahramani (2005) show that the distribution over the equivalence classes of the matrices generated by this model and the IBP are the same, which we briefly present below. See the referred paper for details of the derivation.

Conditioned on μ_k, the rows of the matrix are assumed to be independent, therefore the joint distribution over a column is

$$P(\mathbf{z}_k \mid \mu_k) = (\mu_k)^{m_{\cdot,k}}(1-\mu_k)^{N-m_{\cdot,k}}, \tag{4.10}$$

where $m_{\cdot,k} = \sum_{i=1}^{N} z_{ik}$, is the number of 1s in column k, i.e. the number of objects that share the kth feature. We can integrate over μ_k using the Dirichlet integral (B.11), and express the joint probability of the column directly in terms of the hyperparameter α:

$$\begin{aligned} P(\mathbf{z}_k) &= \int P(\mathbf{z}_k \mid \mu_k) P(\mu_k)\, \mathrm{d}\mu_k \\ &= \int \frac{\Gamma(\alpha/K+1)}{\Gamma(\alpha/K)} \mu_k^{\alpha/K-1} \mu_k^{m_{\cdot,k}} (1-\mu_k)^{N-m_{\cdot,k}}\, \mathrm{d}\mu_k \\ &= \frac{\alpha}{K} \frac{\Gamma(m_{\cdot,k}+\alpha/K)\Gamma(N-m_{\cdot,k}+1)}{\Gamma(N+\alpha/K+1)}. \end{aligned} \tag{4.11}$$

Assuming the columns to be independent, the distribution over the whole matrix becomes

$$P(Z) = \prod_{k=1}^{K} P(\mathbf{z}_k) = \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_{\cdot,k}+\alpha/K)\Gamma(N-m_{\cdot,k}+1)}{\Gamma(N+\alpha/K+1)}. \tag{4.12}$$

Considering the *lof*-equivalence classes of the matrices, the number of matrices Z defined by the above generative model that map to the equivalent matrix $[Z]$ is

$$\frac{K!}{\prod_{h=1}^{2^N-1} K_h!},$$

which leads to the following distribution over $[Z]$,

$$P([Z]) = \frac{K!}{\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_{\cdot,k}+\alpha/K)\Gamma(N-m_{\cdot,k}+1)}{\Gamma(N+\alpha/K+1)}. \tag{4.13}$$

This is the distribution of the equivalence classes for the binary matrix with $K < \infty$ columns. The distribution over the matrix with infinitely many columns can be obtained by separating the terms for non-zero columns (i.e., columns with $m_{\cdot,k} > 0$) and the zero columns and simply taking the limit as the number of columns $K \to \infty$. The limiting distribution is found to be equal to eq. (4.2).

We repeat the argument of the previous section: even though Z has infinitely many columns, its distribution favors sparsity. We can compute the expected number of non-zero entries in Z for the finite case, and take the limit to obtain the expression for the

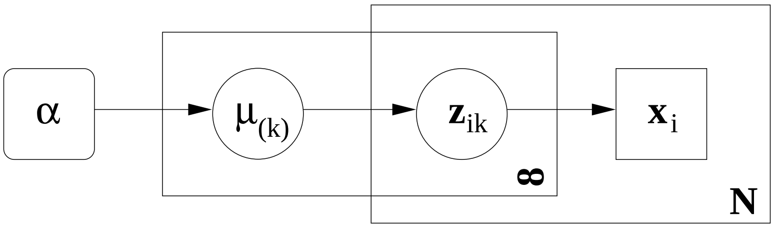

Figure 4.3: Graphical representation for the infinite latent feature model using the stick breaking construction of the Indian buffet process. Note the strictly decreasing order of the feature presence probabilities.

infinite matrix:

$$\begin{aligned}\mathrm{E}\{\mathbf{1}^T Z^T \mathbf{1}\} &= \mathrm{E}\big\{\sum_{ik} z_{ik}\big\} = K\sum_{i=1}^{N}\mathrm{E}\{z_{ik}\} \\ &= K\sum_{i=1}^{N}\int_0^1 \mu_k p(\mu_k)\mathrm{d}\mu_k \\ &= KN\frac{\alpha/K}{1+\alpha/K} = N\frac{\alpha}{1+\alpha/K}\end{aligned} \tag{4.14}$$

Taking the limit as $K \to \infty$, the expected number of non-zero entries for the infinite matrix is found to be αN, consistent with the result of the previous section.

The derivation presented above starts with defining the distribution over a finite dimensional matrix and uses the conjugacy of the beta distribution to the binomial distribution to integrate out the feature presence probabilities μ_k. Although the rows are independent given μ_k, marginalization couples the rows, and the probability of an entry is given in terms of the hyperparameter α and the number of other entries in that column that are set to one, i.e. the number of objects sharing the feature. For the distribution over the infinite matrix to be well defined, the column indices are ignored by focusing on the permutation-invariant equivalence classes of matrices using the *lof*$(\cdot)$ function before taking the infinite limit.

In the next section, we derive the same distribution with another approach, in which the feature presence probabilities are explicitly represented instead of being integrated out. To have a well defined distribution over the matrix with infinitely many columns, a strictly decreasing ordering of the μ_ks is imposed, which results in the so called stick-breaking construction.

4.1.2 Stick-Breaking Construction

In Section 3.1.4 we described the stick-breaking construction for the DP. In this section, we describe a similar construction for the IBP which allows representing the feature presence probabilities explicitly in the IBLF model, see Figure 4.3. The derivation starts from the distribution on a finite binary matrix and imposing a strictly decreasing ordering of the feature presence probabilities before taking the infinite limit. Denoting the kth largest feature presence probability with $\mu_{(k)}$, we will show that the stick-breaking construction for the IBP has the following law for the feature presence probabilities,

$$\mu_{(k)} = \nu_k \, \mu_{(k-1)} = \prod_{l=1}^{k} \nu_l, \quad \text{with} \quad \mu_{(0)} = 1 \quad \text{and} \quad \nu_k \sim \text{Beta}(\alpha, 1). \tag{4.15}$$

That is, each feature presence probability $\mu_{(k)}$ is a product of the previous one, $\mu_{(k-1)}$, and the random variable ν_k. This can be described metaphorically as starting with a stick of unit length ($\mu_{(0)} = 1$), and breaking a piece off the stick at each iteration, discarding that piece and recursing on the piece that we kept. The breaking point is determined by the variable ν_k, and the feature presence probability $\mu_{(k)}$ is the stick length left after k iterations. Note that this procedure enforces a strictly decreasing ordering of the μ's since $\mu_{(k)}$ is given by the stick length that we have left after the kth iteration[1] and we recurse on the piece of the stick left at each iteration, see Figure 4.4 for a pictorial representation.

In the following, we present the outline of derivation of the stick-breaking construction for the IBP. Details of the derivation are given in Appendix A. We start with the same distributional assumptions on the entries of a finite binary matrix with N rows and K columns as in Section 4.1.1. That is, for the unordered columns we assume

$$\begin{aligned} (z_{ik} \,|\, \mu_k) \sim & \text{Bernoulli}(\mu_k) \\ \mu_k \sim & \text{Beta}(\frac{\alpha}{K}, 1). \end{aligned} \tag{4.16}$$

The density of μ_k is given as:

$$\begin{aligned} p(\mu_k) &= \frac{\Gamma(\frac{\alpha}{K}+1)}{\Gamma(\frac{\alpha}{K})\Gamma(1)} \mu_k^{\frac{\alpha}{K}-1} (1-\mu_k)^{1-1} \mathbb{I}\{0 \le \mu_k \le 1\} \\ &= \frac{\alpha}{K} \mu^{\frac{\alpha}{K}-1} \mathbb{I}\{0 \le \mu_k \le 1\}, \end{aligned} \tag{4.17}$$

where $\mathbb{I}\{A\}$ is the indicator function for a measurable set A; $\mathbb{I}\{A\} = 1$ if A is true, 0 otherwise. The cumulative distribution function (cdf) for μ_k is:

$$\begin{aligned} F(\mu_k) &= \int_0^{\mu_k} \frac{\alpha}{K} t^{\frac{\alpha}{K}-1} \mathbb{I}\{0 \le t \le 1\} \mathrm{d}t \\ &= \mu_k^{\frac{\alpha}{K}} \mathbb{I}\{0 \le \mu_k \le 1\} + \mathbb{I}\{1 < \mu_k\}. \end{aligned} \tag{4.18}$$

We define $\mu_{(1)} \ge \mu_{(2)} \ge \cdots \ge \mu_{(K)}$ to be the decreasing ordering of $\mu_1, \ldots, \mu_K$. Thus, $\mu_{(1)}$ is defined as

$$\mu_{(1)} = \max_{k=1,\ldots,K} \mu_k, \tag{4.19}$$

[1] For the DP, the mixing proportions correspond to the length of the piece that we discard. This relation be discussed further in the section

The cumulative distribution function of the maxima of a set of independent random variables is the product of the cumulative distribution functions of each of the variables. We obtain the cdf for $\mu_{(1)}$ by taking the product of the K (identical) cdf's,

$$\begin{aligned} F(\mu_{(1)}) &= \left[\mu_{(1)}^{\frac{\alpha}{K}}\,\mathbb{I}\{0 \le \mu_{(1)} \le 1\} + \mathbb{I}\{1 < \mu_{(1)}\}\right]^K \\ &= \mu_{(1)}^{\alpha}\,\mathbb{I}\{0 \le \mu_{(1)} \le 1\} + \mathbb{I}\{1 < \mu_{(1)}\}. \end{aligned} \tag{4.20}$$

Differentiating, we obtain the probability density function (pdf) of $\mu_{(1)}$:

$$p(\mu_{(1)}) = \alpha \mu_{(1)}^{\alpha-1}\,\mathbb{I}\{0 \le \mu_{(1)} \le 1\}. \tag{4.21}$$

That is, $\mu_{(1)} \sim \text{Beta}(\alpha, 1)$.

Following the same approach for the subsequent feature presence probabilities, the density of $\mu_{(k+1)}$ is obtained to be,

$$\begin{aligned} p(\mu_{(k+1)} \,|\, \mu_{(1:k)}) &= p(\mu_{(k+1)} \,|\, \mu_{(k)}) \\ &= \alpha \frac{K-k}{K} \mu_{(k)}^{-\alpha\frac{K-k}{K}}\, \mu_{(k+1)}^{\alpha\frac{K-k}{K}-1} \mathbb{I}\{0 \le \mu_{(k+1)} \le \mu_{(k)}\}. \end{aligned} \tag{4.22}$$

This is the density of the $(k+1)$th largest value of K random variables all with distribution given in eq. (4.16). To obtain the distribution of the probabilities corresponding to the columns of the infinite matrix, we take the limit as $K \to \infty$. In the limit, the density of $\mu_{(k+1)}$ becomes

$$p(\mu_{(k+1)} \,|\, \mu_{(k)}) = \alpha \mu_{(k)}^{-\alpha}\, \mu_{(k+1)}^{\alpha-1}\, \mathbb{I}\{0 \le \mu_{(k+1)} \le \mu_{(k)}\}. \tag{4.23}$$

Defining $\mu_{(0)} = 1$, the above equation gives the densities for the feature presence probabilities for all columns of the infinite dimensional binary matrix Z sorted in a strictly decreasing order.

Note that given $\mu_{(k)}$, $\mu_{(k+1)}$ is independent of all other μ values. We introduce a set of variables $\nu_k = \frac{\mu_{(k)}}{\mu_{(k-1)}}$ to make use of this Markov property. Since $\mu_{(k)}$ has range $[0, \mu_{(k-1)}]$, ν_k has range $[0, 1]$. Using a change of variables, the distribution of ν_k can be obtained from eq. (4.23) to be,

$$\begin{aligned} p(\nu_k \,|\, \mu_{(k-1)}) &= p(\mu_{(k)} \,|\, \mu_{(1:k-1)}) \left| \frac{\mathrm{d}\mu_{(k)}}{\mathrm{d}\nu_k} \right| \\ &= \alpha \nu_k^{\alpha-1} \mathbb{I}\{0 \le \nu_k \le 1\}. \end{aligned} \tag{4.24}$$

Thus, ν_k is independent from $\mu_{(1:k-1)}$ and is simply $\text{Beta}(\alpha, 1)$ distributed. We obtain the stick-breaking representation by expanding $\mu_{(k)}$,

$$\mu_{(k)} = \nu_k\, \mu_{(k-1)} = \prod_{l=1}^{k} \nu_l. \tag{4.25}$$

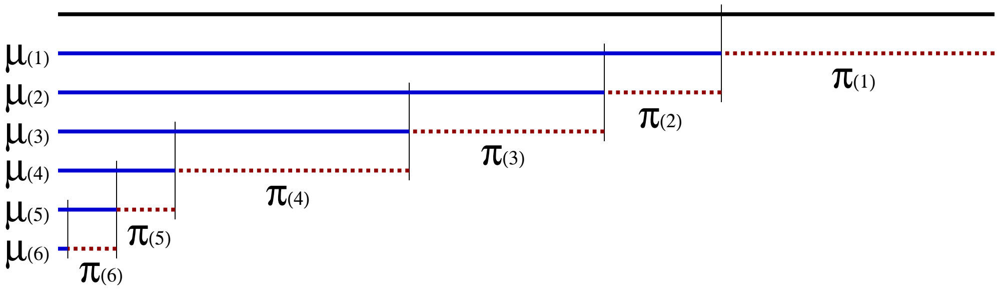

Figure 4.4: Stick-breaking construction for the DP and IBP. The black line at top is the stick of unit length. The vertical lines show the break point of the stick. The horizontal blue lines show the length of the stick left after each iteration, which determines the values for the feature presence probabilities $\mu_{(k)}$. The red dotted lines show the discarded pieces of the stick corresponding to the mixing proportions for the DP.

We refer to this representation as the stick-breaking construction due to its correspondence to the standard stick-breaking construction for the DP (Sethuraman, 1994; Ishwaran and James, 2001).

Let π_k be the length of stick piece discarded at iteration k. We have,

$$\pi_k = (1 - \nu_k)\mu_{(k-1)} = (1 - \nu_k)\prod_{l=1}^{k-1} \nu_l. \tag{4.26}$$

Making a change of variables $v_k = 1 - \nu_k$,

$$v_k \overset{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha), \quad \pi_k = v_k \prod_{l=1}^{k-1}(1 - v_l) \tag{4.27}$$

giving the standard stick-breaking construction for DP, eq. (3.16).

In both constructions the weights of interest are given in terms of the stick lengths. In DP, the weights π_k are the lengths of the discarded pieces, while in IBP, the weights $\mu_{(k)}$ are the length of the stick we keep, see Figure 4.4. This difference leads to the different properties of the weights: for DP, the stick lengths sum to a length of 1 and are not strictly decreasing, but are in a size-biased order. In IBP the stick lengths may sum to a value greater than 1 but are decreasing. For both, the weights decrease exponentially quickly in expectation.

The particular ordering of the columns results in some interesting properties of the distribution on Z. As stated earlier, $\mu_{(k)}$ is the prior probability of any of the objects having feature k. There is a positive probability for any data point to have any of the infinitely many features. However, since the feature presence probabilities have a strictly decreasing order with exponential rate, in expectation the infinite dimensional Z matrix has only finitely many non-zero entries.

We can calculate the distribution of the number of non-zero entries to the right of column k in row i, given $\mu_{(k)}$. This does not depend on i due to the exchangeability of the rows. The distribution of the unordered feature presence probabilities μ_l after the kth largest one is given by eq. (A.2). The entries z_{il} are Bernoulli distributed with probability μ_l. Marginalizing over μ_l, we have;

$$\begin{aligned} p(z_{il} \,|\, \mu_{(k)}) &= \int_0^{\mu_(k)} p(z_{il} \,|\, \mu_l) p(\mu_l \,|\, \mu_{(k)}) \mathrm{d}\mu_l \\ &= \int_0^{\mu_(k)} \mu_l \frac{\alpha}{K} \mu_{(k)}^{-\alpha/K} \mu_l^{\alpha/K-1} \mathrm{d}\mu_l \\ &= \frac{\alpha}{\alpha+K} \mu_{(k)}. \end{aligned} \tag{4.28}$$

Taking the limit as $K \to \infty$ of the Bernoulli trials with the above probability results in a Poisson$(\alpha\mu_{(k)})$ distribution over the number of non-zero entries in the ith row. This result is intuitive, stating that the expected number of non-zero entries depends on the limiting $\mu_{(k)}$ value, and the parameter α. If we consider the entries of the whole row, i.e. the case $k = 0$, we recover the distribution derived in the previous section for the total number of non-zero entries in a row, Poisson(α).

We can also calculate the probability of all entries to the right of column k being zero:

$$p(Z_{(:,>k)} = 0 \,|\, \mu_{(k)}) = \exp\big(-\alpha H_N + \alpha \sum_{i=1}^{N} \frac{(1-\mu_{(k)})^i}{i} \big), \tag{4.29}$$

where H_N is the Nth harmonic number. See Appendix A for the details of derivation.

4.1.3 A Special Case of the Beta Process

Beta process has been defined for use in survival analysis as a cumulative hazard rate process with nonnegative independent increments[2] by Hjort (1990). Thibaux and Jordan (2007) show that a special case of the beta process is related to the Indian buffet process (IBP) the way the Dirichlet process is related to the Chinese restaurant process.

They define the following generative model for the binary feature matrix Z:

$$\begin{aligned} A &\sim \mathrm{BP}(c, \alpha A_0), \\ Z &\sim \mathrm{BeP}(A), \end{aligned} \tag{4.30}$$

where BeP(A) denotes a Bernoulli process with hazard measure A and BP$(c, \alpha A_0)$ denotes a beta process with base measure αA_0 and concentration function c. A_0 is a probability measure, α a positive scalar, and c a deterministic function. The distribution of Z is equivalent to that of the matrix generated by the IBP(α) for the particular choice of c being the constant function $c = 1$, BP$(1, \alpha A_0)$.

Considering the beta process BP$(c, \alpha A_0)$ with a constant concentration function in the above generative model results in a two-parameter generalization of the Indian buffet

[2] See Appendix B.3 for the definition of the independent increment processes

process. Customer i samples the previously sampled dishes with Bernoulli probability $\left(\frac{m_{\cdot<i,k}}{c+i}\right)$ and samples Poisson$(\frac{c\alpha}{c+i})$ new dishes. We can sequentially generate from this generalized Indian buffet process for N customers. In detail, at each step (for each customer) $i \geq 1$,

– Sample the number of new atoms $K^{(i)} \sim$ Poisson$(\frac{c\alpha}{c+i})$
– Sample $K^{(i)}$ new locations θ_j independently from A_0
– Sample their weight $\mu_j \sim$ Beta$(1, c+i-1)$ independently
– Define $A_i = A_{i-1} + \sum_{j=1}^{K^{(i)}} \mu_j \delta_{\theta_j}(\cdot)$

The resulting A_N is an approximation of the beta process A since $\lim_{N\to\infty} A_N = A$ with probability one. Note that the weights μ_j of the atoms are the feature presence probabilities defined in the previous sections. In this construction, they are generated in a size-biased order, similar to the weights π_j of the stick-breaking construction of the DP.

Beta process is an independent increment process with the corresponding Lévy measure given by

$$dN(\omega, x) = c(\omega)x^{-1}(1-x)^{c(\omega)-1}\mathrm{d}x\, A_0(\mathrm{d}\omega), \tag{4.31}$$

(Hjort, 1990). Therefore, it can be represented as a countably infinite sum of random jumps at random points, similar to the gamma process discussed in Section 3.1.3. Unlike the gamma process, the beta process has fixed points of discontinuity, i.e. atoms at fixed locations but with random mass. Wolpert and Ickstadt (1998) show how to construct the beta process using the inverse Lévy measure (ILM) which results in the strictly decreasing ordering of the feature presence probabilities. Thus, the stick-breaking construction for the IBP presented in Section 4.1.2 turns out to be an elegant way of describing the ILM for the beta processes with a constant concentration function.

4.1.4 Properties of the Distribution on the Infinite Binary Martices

We have summarized the different ways of describing the distribution on the binary matrices imposed by the IBP, and the properties of the distribution. Here, we give an overview of the properties that are important for building IBLF models and developing techniques for inference.

The IBP produces binary matrices with infinitely many columns. The rows of the matrix can be thought of as corresponding to the observations and the columns to the binary features of the observations. An entry being 1 means the presence and 0 means the absence of a feature for an observation. The key idea is that the features are not mutually exclusive. That is, each observation can have any number of the infinitely many features. Consequently, the total number of active features (nonzero columns) for a certain number of observations (rows) is not limited *a priori*. One of the most important properties of IBP that makes inference tractable is its sparsity. Only a small finite number of columns will have nonzero entries. Defining the model such that only the active feature columns play a role in the likelihood, we do not have to deal with the infinitely many inactive columns. For a matrix with N rows, the distribution of the number of nonzero columns follows a Poisson(αH_N) distribution. Each row has

Poisson(α) nonzero entries, and the distribution of the total nonzero entries in the matrix follows a Poisson(αN) distribution.

The equivalence classes have been defined by using the *lof*(.) function. The ordering of the columns is not important for the *lof* representation, however the stick-breaking representation has a particular ordering. The columns are sorted with decreasing feature presence probabilities $\mu_{(k)}$. This representation allows the $\mu_{(k)}$ being represented in computations rather than being integrated out. The construction of the feature presence probabilities is such that the rate of decay is exponential. Given a part of the matrix, this representation allows judgment about the distribution of entries of the rest of the matrix. In particular, after a column with a feature presence probability $\mu_{(k)}$, the expected number of nonzero entries in the rest of the row has a Poisson($\alpha\mu_{(k)}$), and the rest of the matrix a Poisson($\alpha\mu_{(k)}N$) distribution.

The stick-breaking construction for the IBP and for the DP has an interesting relation. For the IBP, the feature presence probabilities $\mu_{(k)}$ correspond to the stick lengths that we have left after breaking a piece off the stick, and for the DP the mixing proportions π_k correspond to the piece we discard. For this reason, the stick-breaking construction for the IBP has strictly decreasing weights and the DP has size-biased ordering of the weights.

The direct correspondence to stick-breaking in DP implies that a range of techniques for and extensions to the DP can be adapted for the IBP. For example, we can generalize the IBP by replacing the Beta($\alpha, 1$) distribution on ν_k's with other distributions. One possibility is a Pitman-Yor extension (Pitman and Yor, 1997) of the IBP, defined as

$$\nu_k \sim \text{Beta}(\alpha + kd, 1 - d) \qquad\qquad \mu_{(k)} = \prod_{l=1}^{k} \nu_l \tag{4.32}$$

where $d \in [0, 1)$ and $\alpha > -d$. The Pitman-Yor IBP weights decrease in expectation as a $O(k^{-\frac{1}{d}})$ power-law that has heavier tails for the distribution of the number features. This may be a better fit for some naturally occurring data which have a larger number of features with significant but small weights Goldwater et al. (2006).

An example technique for the DP which we can adapt to the IBP is to truncate the stick-breaking construction after a certain number of break points and to perform inference in the reduced space. Ishwaran and James (2001) give a bound for the error introduced by the truncation in the DP case which can be used here as well.

Thibaux and Jordan (2007) show the connection between the beta process and the IBP. This suggests that the beta process, which has been extensively used in survival analysis, can be used for defining binary latent feature models. Furthermore, they use the connection between the IBP and the beta process to develop a new algorithm to sample beta processes with size-biased ordering of the weights.

Wolpert and Ickstadt (1998) show how to construct the beta process with strictly decreasing weights using the inverse Lévy measure. The stick-breaking construction is a neat way of describing this algorithm.

A direct consequence of the stick-breaking construction and the relation of the IBP to the beta process is that a draw from a beta process has the form $\sum_{k=1}^{\infty} \mu_{(k)} \delta_{\theta_k}(\cdot)$ with $\mu_{(k)}$ drawn form the stick breaking prior for the feature presence probabilities, and $\theta_{(k)}$ is independent from $\mu_{(k)}$ and is drawn from the base measure H. Generalizations of the stick-breaking constructions lead to generalizations of the beta process.

The IBP parameter α affects the number of active features therefore updating this parameter would give more flexibility to the model. The likelihood for α can be derived from the joint distribution of the features given in Equation 34 of Griffiths and Ghahramani (2005) to be,

$$p(Z \mid \alpha) \propto \alpha^{K^{\ddagger}} \exp\big(- \alpha H_N \big),$$

where $K^{\ddagger}$ is the number of active components and N is the number of rows. We can put a gamma prior on α,

$$\alpha \sim \mathcal{G}(1, 1).$$

Combining this likelihood with the prior, we get the posterior distribution for α

$$p(\alpha \mid Z) = \mathcal{G}\big(1 + K^{\ddagger}, 1 + H_N\big). \tag{4.33}$$

Since this posterior is of standard form, it can be sampled from easily.

Note that the construction of IBP and related distributions described above concerns the prior for Z only. Using Z in a IBLF, we update it by sampling its entries using the full conditional distributions which involves conditioning on the data. In the following section, we describe methods for inference on IBLF models.

4.2 MCMC Sampling algorithms for IBLF models

The previous section summarized several different approaches for defining the distribution induced by the Indian buffet process, which has strong connections to the Dirichlet Process. In this section, we describe MCMC methods for inference on the latent feature models that use the IBP as the nonparametric prior over the feature matrix, which we refer to as the infinite binary latent feature (IBLF) models.

The general form of the models we will consider assumes the data generating process to be living in a latent space of possibly infinite dimension. Each data point $\mathbf{x}_i$ is described by an infinite dimensional binary latent vector $\mathbf{z}_i = z_{i,1:\infty}$ that encodes the presence or absence of the infinitely many features characterized by the parameters $\Theta = \theta_{1:\infty}$, and possibly other parameters Φ that live in the observable space. The state of the binary latent variables determine which features are actively used, that is, the effective latent dimensionality. Equivalently, we refer to the vector $\mathbf{z}_i$ as the binary latent feature vector, and Θ as the set of parameters of the features. We put an IBP prior on the feature presence matrix Z, which has the vectors $\mathbf{z}_i$ as its rows. The parameters Θ and Φ are given standard parametric priors. The data distribution is assumed not to depend on the features that do not belong to any of the data points, that is, the zero columns of Z and the parameters θ_k that are associated with those columns. The model can be

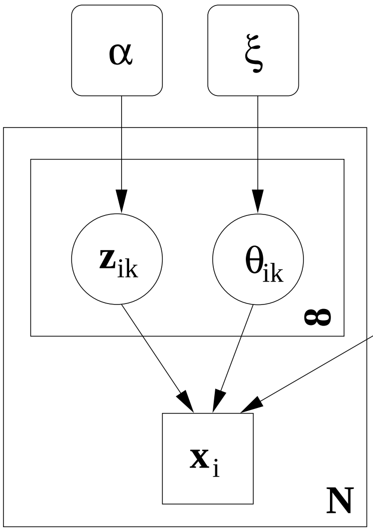

Figure 4.5: Graphical representations of a general latent feature model. The data $\mathbf{x}_i$ is generated by latent features, $\mathbf{z}_i$, parameters associated with the latent features, θ_k and other parameters ϕ, all of which have assigned prior distributions. The hierarchy can be extended by specifying priors on the hyperparameters.

summarized as follows:

$$\begin{aligned}(\mathbf{x}_i \,|\, \mathbf{z}_i, \Theta, \Phi) &\sim F(\mathbf{x}_i \,|\, \mathbf{z}_i, \Theta, \Phi)\\ Z &\sim IBP(\alpha)\\ \theta_k &\sim G_0(\theta_k \,|\, \xi)\\ \Phi &\sim H(\Phi \,|\, \gamma)\end{aligned} \tag{4.34}$$

where $F(\cdot)$ is the distribution of the data and G_0 and H are prior distributions for the parameters, specified by hyperparameters. See Figure 4.5 for a graphical representation. The model can be extended by adding more layers to the hierarchy by specifying priors for the hyperparameters. Note that there are infinitely many latent features. But since the data distribution does not depend on the inactive features, inference on this model is still tractable.

We will consider updating each set of variables sequentially. Given the nonparametric part of the model concerning the latent variables, i.e. given Z and Θ, the updates for the rest of the parameters Φ is just like in the parametric hierarchical models. Furthermore, although Θ is infinite dimensional, Z will only have finitely many non-zero columns. Since the likelihood does not depend on the zero columns, the posterior for θ_k corresponding to the inactive columns of Z is same as the prior. Therefore, we only need to update those θ_k that are associated with the *active* features. This is again simply sampling from the conditional posterior of θ_k,

$$P(\theta_k \,|\, X, Z, \Phi) \propto G_0(\theta_k \,|\, \xi) F(X \,|\, Z, \Theta, \Phi) \tag{4.35}$$

as in the parametric models. Updating Z is more involved, since we have to deal with the infinite dimensionality. Therefore, in the rest of this section, we will focus on updates concerning Z.

In the following, we describe several different MCMC algorithms for inference on the IBLF models. Conjugacy of the distribution of the feature parameters to the likelihood function makes inference computationally easier. However, requiring conjugacy limits the use of IBP. We start with describing sampling for conjugate IBLF models and continue with other sampling algorithms that do not require conjugacy.

As for the DP, the sampling algorithms for IBLF models may be divided into two subgroups: the ones that integrate out the feature presence probabilities μ_k, and the ones that explicitly represent these probabilities using the stick-breaking construction. We present algorithms that use these different approaches in the following subsections.

All methods described below update each entry of Z incrementally. Note that in the IBP, the customers use different criteria for choosing dishes already sampled and for choosing new dishes. As a consequence, the methods that use the representation where μ_k is integrated out employ different update rules for the features owned by other data points and for the (existing or new) unique features of the data point being considered. On the other hand, for the methods that use the stick-breaking construction, we need not make such a distinction.

In the following sections, Z denotes the binary matrix (with unbounded number of columns), z_{ik} an entry of Z. The index $i = 1, \ldots, N$ runs over the rows, and $k = 1, \ldots, K$ runs over the columns of Z. Similar to the terminology used in the previous chapter for the DPM models, the columns of Z with non-zero entries will be referred to as the active features, and the columns of zeros as the inactive features. Although Z has infinitely many columns, only finitely many of them will be active. We denote the number of active features with $K^{\ddagger}$ and the number of features explicitly represented for the computations (which includes all active features and possibly some inactive features) with $K^{\dagger}$. The variable m is used to denote the number of entries that are set to 1 in a specified part of Z, e.g. $m_{\cdot,k}$ means number of 1s in the kth column, and $m_{\cdot<i,k}$ the same for the kth column for rows $1, \ldots, i-1$.

4.2.1 Gibbs Sampling for Conjugate IBLF models

Griffiths and Ghahramani (2005) show how to do inference using Gibbs sampling for the IBLF models with conjugate priors. If the prior distribution for Θ is conjugate to the likelihood, we can update the model parameters and the entries of the binary features by sampling from their full conditionals. Algorithm 11 summarizes this method.

In detail, for updating the entries of Z, we can exploit the exchangeability of the rows of the matrix and treat the data point being considered as the last data point. We use $m_{-i,k}$ to denote the number of data points other than i that have feature k. The conditional posterior for the features with $m_{-i,k} > 0$ is proportional to the product of the prior given in eq. (4.7) and the likelihood;

$$p(z_{ik} = 1 \mid Z_{-ik}, X, \Theta, \Phi) \propto \frac{m_{-i,k}}{N} F(X \mid z_{ik} = 1, Z_{-ik}, \Theta, \Phi). \tag{4.36}$$

For the features for which $m_{-i,k} = 0$, i.e, the features that no object has (other than possibly object i), the prior probability approaches zero when we consider $K \to \infty$. Therefore, instead of considerig these features separately, we consider the distribution over the *number* of new features $K_*^{(i)}$ for the ith data point[3] which has a Poisson(α/N) prior. There is a parameter θ_k associated with each column $k = 1, \ldots, \infty$ of the latent feature matrix Z. To obtain the posterior distribution over the number of features, we need to integrate over the parameters to get the marginal likelihood,

$$F(X \mid Z, \Phi) = \int F(X \mid Z, \Theta, \Phi) \mathrm{d}G_0(\Theta \mid \xi). \tag{4.37}$$

This integral is analytically tractable only for the case where the prior for Θ is conjugate to the likelihood function $F(X \mid Z, \Theta, \Phi)$. The conjugacy requirement is similar to the DP models. Without conjugacy, we need to represent the parameters associated with each feature. Therefore the features are no longer exchangeable and we need to choose *which* of the infinitely many features to choose. When we integrate over the parameters, exchangeability is preserved and we only need to decide *how many* features to include.

Combining the Poisson prior with the marginal likelihood gives the conditional posterior;

$$\begin{aligned} p(K_*^{(i)} \mid Z_{-K^{(i)}}, X, \Phi) &\propto \frac{(\alpha/N)^{K_*^{(i)}} \exp(-\alpha/N)}{K_*^{(i)}!} \int F(X \mid Z_{K_*^{(i)}}, Z_{-K^{(i)}}, \Theta, \Phi) \mathrm{d}P(\Theta) \\ &\propto \frac{(\alpha/N)^{K_*^{(i)}} \exp(-\alpha/N)}{K_*^{(i)}!} F(X \mid Z_{K_*^{(i)}}, Z_{-K^{(i)}}, \Phi), \end{aligned} \tag{4.38}$$

where $Z_{-K^{(i)}}$ denotes the part of the matrix Z without the columns corresponding to the *existing* unique features of data point i, and $Z_{K_*^{(i)}}$ denotes the columns with the $K_*^{(i)}$ *new* unique features for i.

Note that, to have a closed form expression for the posterior eq. (4.38), we also need the marginal likelihood function $F(X \mid Z, \Phi)$ to be conjugate to the Poisson prior. If this is not the case, we can sample from the posterior for example by approximating the posterior by evaluating the probabilities for a range of values $K_*^{(i)}$ up to an upper bound.

Although the application of Gibbs sampling to the conjugate models is straightforward, requiring conjugacy limits the models that can be developed using IBP. In the following, we describe algorithms that do not require conjugacy.

[3] We overload the notation here. Note that in the previous section, $K_*^{(i)}$ was used also as the number of new features for object i, but here we assume each object to be the last since we condition on all others. Here, the superscript i denotes the identity of the data point, not its place in the sequence.

Algorithm 11 Gibbs sampling for conjugate IBP

```
The state of the Markov chain consists of the infinite feature matrix $Z$ and the set of infinitely many parameters $\Theta = \theta_{1:\infty}$.
Only the $K^{\ddagger}$ active columns of $Z$ and the corresponding parameters are represented.
Repeatedly sample:
for all rows $i = 1, \ldots, N$ do {Feature updates}
  for all columns $k = 1, \ldots, K^{\ddagger}$ do
    if $m_{-i,k} > 0$ then
      Update $z_{ik}$ by sampling from its conditional posterior, eq. (4.36).
    end if
    Remove the columns of $Z$ for which $m_{-i,k} = 0$ and the corresponding parameters $\theta_k$ from the representation.
    Sample the number of new unique features for $i$ from the posterior, eq. (4.38).
    Update $Z$ to include the new number of unique features of $i$.
  end for
end for
for all active columns $k = 1, \ldots, K^{\ddagger}$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35).
end for
```

4.2.2 Approximate Gibbs Sampling for Non-Conjugate Models

For the models in which the prior for the parameters Θ and the likelihood are not conjugate, the likelihood cannot be marginalized over Θ. Hence, the parameters associated also with each inactive feature column need to be represented explicitly. This prevents us from computing the limiting posterior distribution over the infinite feature matrix. The prior for the entries of the columns with $m_{-i,k} > 0$ is well defined, therefore we can still use Gibbs sampling for updating these features, conditioning also on the parameter values. For sampling for the unique features, Görür et al. (2006) propose using an approximation to the posterior.

Note that in Section 4.1.1, the Poisson prior over the number of new features has been obtained as the limit of K Bernoulli trials with probability $\frac{\alpha/K}{N+\alpha/K}$ which approaches zero as $K \to \infty$. We can obtain an approximation to the infinite case by truncating the number of Bernoulli trials at a finite value $\hat{K}$, and considering the joint posterior probability of these $\hat{K}$ features. Note the difference compared to a finite model where the number of total features is fixed at a certain value. There are always infinitely many features, most of them being inactive, hence they do not affect the likelihood. However, when doing posterior updates, we need to take into account the possibility of any of these features becoming active. We use the truncation as an approximation for the Gibbs sampling updates only for the (existing or new) unique features. Therefore, the number of features the model can introduce is not bounded, contrary to a model with a finite but large number of features.

We use $\hat{K}$ auxiliary variables to represent the possible values for the parameters $\hat{\Theta} = \hat{\theta}_{1:\hat{K}}$ of the $\hat{K}$ features that are not associated with any other object. We associate parameters of the existing unique features of object i with some of the auxiliary parameters and draw values from the prior $G_0(\hat{\theta}_j \mid \xi)$ for the rest of the auxiliary parameters. Since each feature has a parameter associated with it, the particular order of the columns is important, unlike the case in the previous section. There are $2^{\hat{K}}$ possible combinations for the $\hat{K}$ unique features for object i. We denote the part of the matrix with active features that are not unique to object i with $Z_{-K^{(i)}}$ and the auxiliary features that are not associated with any object other than (possibly) object i with $\hat{Z}_l$ where the index $l \in [1, \ldots, 2^{\hat{K}}]$ denotes the possible feature settings. (Note that all entries of $\hat{Z}_l$ are zero except the ith row.) We evaluate the posterior probabilities of all possible $\hat{Z}_l$ and sample from this distribution to decide on which to include. The joint posterior for the ith row of $\hat{Z}_l$ will consist of the prior Bernoulli probabilities of setting each feature in the ith row to 0 or 1 (with probability $\frac{\alpha/\hat{K}}{N+\alpha/\hat{K}}$), and the probability of the data given $Z_{-K^{(i)}}$, $\theta_{-K^{(i)}}$, $\hat{Z}_l$ and $\hat{\Theta}$,

$$P(\hat{Z}_l \mid X, Z_{-K^{(i)}}, \Theta_{-K^{(i)}}, \hat{\Theta}) \propto P\left(\hat{Z}_l\right) P\left(X \mid \hat{Z}_l, Z_{-K^{(i)}}, \Theta_{-K^{(i)}}, \hat{\Theta}\right). \tag{4.39}$$

The IBLF models have close correspondences with the DPM models which allows infinite components in the mixture model. The unique features of an object can be thought of as the singleton components in the DP, and the inactive features as the mixture components that do not have any data associated with them. The sampling scheme described in this section (summarized as Algorithm 12) is inspired from "Algorithm 8" of Neal (2000) for inference in the Dirichlet process models with non-conjugate priors (described in Section 3.2.2, Algorithm 5). However, note that Neal's algorithm for DP models is exact, whereas the algorithm described here for the IBLF models uses an approximation. The quality of approximation gets better with larger values of truncation, however it should be noted that the stationary distribution will be different than the actual posterior distribution due to the approximation.

4.2.3 Metropolis-Hastings Sampling

As discussed above, not having conjugacy is a problem only when Gibbs sampling for the unique features of the data point being considered. Meeds et al. (2007) suggest using Gibbs sampling for the features with $m_{-i,k>0}$ and treating the unique features separately using Metropolis-Hastings sampling, which results in the true posterior distribution. We describe the algorithm below and give a summary in Algorithm 13.

The set of features with $m_{-i,k} = 0$ contains the finitely many unique features of data point i and the infinitely many features that do not belong to any of the data points. Instead of calculating the posterior over the number of new features and sampling directly from this distribution as in Section 4.2.1, we can propose the number of unique features $\hat{K}$ and the set of parameters associated with them $\hat{\Theta} = \hat{\theta}_{1:\hat{K}}$ from a proposal distribution $Q(\hat{K}, \hat{\Theta})$ and consider this proposal with a Metropolis Hastings acceptance

Algorithm 12 Approximate Gibbs sampling for non-conjugate IBP

```
The state of the Markov chain consists of the infinite feature matrix $Z$ and the set of
infinitely many parameters $\Theta = \theta_{1:\infty}$.
Only the $K^\ddagger$ active columns of $Z$ and the corresponding parameters are represented.
Repeatedly sample:
for all rows $i = 1, \ldots, N$ do {Feature updates}
  for all columns $k = 1, \ldots, K^\ddagger$ do
    if $m_{-i,k} > 0$ then
      Update $z_{ik}$ by sampling from its conditional posterior, eq. (4.36).
    else {$m_{-i,k} = 0$}
      Set one of the unallocated auxiliary parameters $\hat{\theta}_j$ to $\theta_k$.
    end if
  end for
  Remove the columns of $Z$ for which $m_{-i,k} = 0$ and the corresponding parameters
  $\theta_k$ from the representation.
  Sample values from the prior $G_0$ for the yet unallocated auxiliary parameters $\hat{\theta}_j$.
  for all $l = 1, \ldots, 2^{\hat{K}}$ do
    Calculate the posterior for $\hat{Z}_l$ using eq. (4.39)
  end for
  Pick a feature combination $\hat{Z}_l$ with probabilities calculated above
  Update $\Theta$ by appending $\hat{\Theta}$
  Update $Z$ by appending $\hat{Z}_l$
  Remove zero columns of $Z$ and the corresponding parameters $\theta_k$ from the representation.
  Discard $\hat{Z}_l$ and $\hat{\Theta}$
end for
for all active columns $k = 1, \ldots, K^\ddagger$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35)
end for
```

ratio.

Meeds et al. (2007) choose the product of the prior distributions as the proposal distribution; $Q(\hat{K}, \hat{\Theta} \,|\, n, \Theta) = \text{Poisson}(\hat{K} \,|\, \alpha/N) G_0(\hat{\Theta} \,|\, \xi)$. That is, the number of unique features $\hat{K}$ for data point i is chosen from the Poisson distribution, and that many parameters are sampled from the prior distribution $G_0(\theta \,|\, \xi)$. We use the same notation as the previous sections and denote the existing unique features of i with $K^{(i)}$. Thus, the feature matrix without these features and the corresponding set of parameters are denoted with $Z_{-K^{(i)}}$ and $\Theta_{-K^{(i)}}$, respectively. The acceptance ratio is given as

$$\begin{aligned}
&\frac{P(\hat{K}, \hat{\Theta} \,|\, X)}{P(K^{(i)}, \Theta \,|\, X)} \frac{Q(K^{(i)}, \Theta \,|\, \hat{K}, \hat{\Theta})}{Q(\hat{K}, \hat{\Theta} \,|\, K^{(i)}, \Theta)} \\
&\qquad = \frac{F(X \,|\, \hat{K}, \Theta_{-K^{(i)}}, \hat{\Theta}, Z_{-K^{(i)}}, \hat{Z}, \Phi)}{F(X \,|\, K^{(i)}, \Theta, Z, \Phi)} \\
&\qquad \times \frac{\text{Poisson}(\hat{K} \,|\, \alpha/N) G_0(\hat{\Theta} \,|\, \xi)}{\text{Poisson}(K^{(i)} \,|\, \alpha/N) G_0(\Theta \,|\, \xi)} \frac{Q(K^{(i)}, \Theta \,|\, \hat{K}, \Theta_{-K^{(i)}}, \hat{\Theta})}{Q(\hat{K}, \Theta_{-K^{(i)}}, \hat{\Theta} \,|\, K^{(i)}, \Theta)} \\
&= \frac{F(X \,|\, \hat{K}, \Theta_{-K^{(i)}}, \hat{\Theta}, Z_{-K^{(i)}}, \hat{Z}, \Phi)}{F(X \,|\, K^{(i)}, \Theta, Z, \Phi)}.
\end{aligned} \tag{4.40}$$

Thus, the acceptance probability reduces to the ratio of the likelihoods. Note that even if the current number of unique features for i is greater than zero, the parameters corresponding to these features are ignored, and all proposal parameters are sampled from the prior. This might lead to low acceptance probabilities if the prior distribution is not a good representation of the posterior distribution for the parameters.

For the conjugate models, the parameters Θ can be integrated out, resulting in the marginal likelihood $F(X \,|\, n, Z, \Phi)$. In that case, we only need to consider sampling the number of unique features to introduce with the simpler proposal distribution $Q(\hat{K} \,|\, n) = \text{Poisson}(\hat{K} \,|\, \alpha/N)$. The acceptance ratio is found to be the ratio of marginal likelihoods:

$$\frac{P(\hat{K} \,|\, X)}{P(\hat{K} \,|\, X)} \frac{Q(\hat{K} \,|\, \hat{K})}{Q(\hat{K} \,|\, \hat{K})} = \frac{F(X \,|\, \hat{K}, Z_{-K^{(i)}}, \hat{Z}, \Phi)}{F(X \,|\, \hat{K}, Z, \Phi)} \tag{4.41}$$

This algorithm has the true posterior distribution as its stationary distribution since it does not suffer from the approximation error introduced by approximating the distribution of the unique features as in the Gibbs sampling algorithm of Section 4.2.2. However, acceptance ratios for non-conjugate models are likely to be small, since we cannot expect the prior to be a good representation of the posterior distribution of the parameters. The parameters of the existing unique features are ignored, and the parameters for the proposed unique features are sampled from the prior.

4.2.4 Gibbs Sampling for the Truncated Stick-Breaking Construction

The stick-breaking construction described in Section 4.1.2 results in a strictly decreasing ordering of the feature presence probabilities. The stick lengths decrease with exponen-

Algorithm 13 Metropolis-Hastings sampling for IBP

```
The state of the Markov chain consists of the infinite feature matrix $Z$ and the set of infinitely many parameters $\Theta = \{\theta_k\}_1^\infty$.
Only the $K^\ddagger$ active columns of $Z$ and the corresponding parameters are represented.

Repeatedly sample:
for all rows $i = 1, \dots, N$ do {Feature updates}
  for all columns $k = 1, \dots, K^\ddagger$ do
    if $m_{-i,k} > 0$ then
      Update $z_{ik}$ by sampling from its conditional posterior, eq. (4.36).
    end if
  end for
  Propose a number $\hat{K}$ for unique features from the prior Poisson($\alpha/N$)
  Sample $\hat{K}$ parameters for the unique features. and $\Theta_{\hat{K}}$
  Evaluate the proposal using the acceptance ratio, eq. (4.40)
end for
for all active columns $k = 1, \dots, K^\ddagger$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35)
end for
```

tial rate, which suggests adapting the truncation used for approximating the DP to the IBP. The bound on the error introduced by the truncation for the DP stick-breaking construction has been calculated by Ishwaran and James (2001). Noting the correspondences between the stick weights of the IBP and the DP, a similar approach can be used in this case.

Let M be the truncation level. Setting $\mu_{(M+1)} = 0$ constrains all $\mu_{(k)} = 0$ for $k > M$, while the joint density for $\mu_{(1:M)}$ is given as

$$\begin{aligned} p(\mu_{(1:M)}) =& \prod_{k=1}^{M} p(\mu_{(k)} \,|\, \mu_{(k-1)}) \\ =& \alpha^M \mu_{(M)}^{\alpha} \prod_{k=1}^{M} \mu_{(k)}^{-1}\, \mathbb{I}\{0 \le \mu_{(M)} \le \cdots \le \mu_{(1)} \le 1\} \end{aligned} \tag{4.42}$$

Inference using Gibbs sampling is straightforward to implement on the truncated model. The entries of Z are independent given $\mu_{(1:M)}$, thus

$$p(Z \,|\, \mu_{(1:M)}) = \prod_{i=1}^{N} \prod_{k=1}^{M} \mu_{(k)}^{z_{ik}} (1 - \mu_{(k)})^{1 - z_{ik}}. \tag{4.43}$$

Since the entries in a column are independent given the feature presence probabilities, we do not need to worry about whether the other data points have the feature being updated or not. That is, we do not need separate update rules for separate cases in this representation. We can sample entries of all columns from their conditional posterior

$$p(z_{ik} \mid \mu_{(k)}, X, \Theta, \Phi) \propto \mu_{(k)} F(X \mid z_{ik} = 1, Z_{-ik}, X, \Theta, \Phi). \tag{4.44}$$

Since the feature presence probabilities are explicitly represented in this construction, they should also be updated. We obtain the posterior for $\mu_{(k)}$ by combining the prior from eq. (4.42) and the likelihood from eq. (4.43),

$$P(\mu_{(k)} \mid Z, \mu_{-(k)}) = \alpha^M \mu_{(M)}^{\alpha} \, \mu_{(k)}^{m_k - 1} \, (1 - \mu_{(k)})^{N - m_k} \, \mathbb{I}\{\mu_{(k+1)} \leq \mu_{(k)} \leq \mu_{(k-1)}\} \tag{4.45}$$

This density is not of standard form. But it can be shown that the log posterior is log-concave, therefore it can be sampled from efficiently using ARS.

Algorithm 14 Gibbs sampling for truncated IBP

The state of the Markov chain consists of the finite feature matrix Z with M columns, the feature presence probabilities $\mu_{1:M} = \mu_1, \ldots, \mu_M$ corresponding to each feature column and the set of parameters $\Theta = \{\theta_k\}_1^M$. All variables are represented.

```
Repeatedly sample:
for all rows $i = 1, \ldots, N$ and columns k=1,...,M do {Feature updates}
  Update $z_{ik}$ by sampling from its conditional posterior, eq. (4.44).
end for
for all columns $k = 1, \ldots, M$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35)
end for
for all columns $k = 1, \ldots, M$ do {Update feature presence probabilities}
  Update $\mu_k$ by sampling from its conditional posterior, eq. (4.45) using ARS
end for
```

4.2.5 Slice Sampling Using the Stick Breaking Construction

Inference on the truncated stick-breaking construction using Gibbs sampling is easy to implement and due to the ordering of the feature presence probabilities, the error introduced by the truncation can be bounded. However, it is possible to avoid approximation all together and do inference on the complete nonparametric model by using slice sampling (Teh, Görür, and Ghahramani, 2007). This method can be interpreted as adaptively choosing a truncation level at each iteration. See (Neal, 2003) for an overview of slice sampling.

Slice sampling has been successfully applied to DP mixture models by Walker (2006) (see Section 3.2.3), and the algorithm for IBLF models described below is related to this approach.

In detail, we introduce an auxiliary slice variable,

$$s \mid Z, \mu_{(1:\infty)} \sim \text{Uniform}[0, \mu^*] \tag{4.46}$$

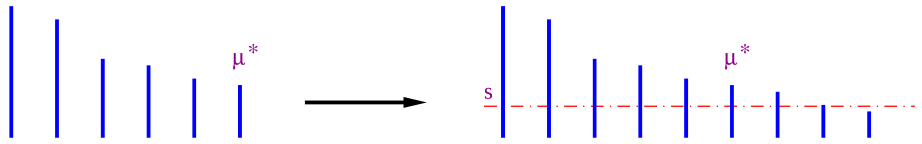

Figure 4.6: Pictorial representation of slice sampling for IBP. The slice value is sampled uniformly randomly between 0 and the stick length of the last active feature. The slice is cut through the stick pieces. The representation is extended to include all features with stick lengths higher than the slice.

where μ^* is a function of $\mu_{(1:\infty)} = \{\mu_{(k)}\}_1^\infty$ and Z, and is chosen to be the length of the stick for the last feature with a non-zero entry,

$$\mu^* = \min\left\{1, \min_{k:\, \exists i, z_{ik}=1} \mu_{(k)}\right\}. \tag{4.47}$$

The joint distribution of Z and the auxiliary variable s is

$$p(Z, s \,|\, \mathbf{x}, \mu_{(1:\infty)}) = p(Z \,|\, \mathbf{x}, \mu_{(1:\infty)})\, p(s \,|\, Z, \mu_{(1:\infty)}) \tag{4.48}$$

where

$$p(s \,|\, Z, \mu_{(1:\infty)}) = \frac{1}{\mu^*}\mathbb{I}\{0 \le s \le \mu^*\}. \tag{4.49}$$

Clearly, integrating out s preserves the original distribution over Z, while conditioned on Z and $\mu_{(1:\infty)}$, s is simply drawn from (4.46). Given s, the distribution of Z becomes:

$$p(Z \,|\, \mathbf{x}, s, \mu_{(1:\infty)}) \propto p(Z \,|\, \mathbf{x}, \mu_{(1:\infty)}) \tfrac{1}{\mu^*}\mathbb{I}\{s \le \mu^*\} \tag{4.50}$$

which forces all columns k of Z for which $\mu_{(k)} < s$ to be zero. This can be interpreted as the auxiliary variable s cutting a slice through the feature presence probabilities, leaving only those with a larger weight than s to be updated. Since s is sampled uniformly between 0 and the height of the last active feature, the features below the slice are fixed to zero.

Let $\tilde{K}$ be the maximal feature index with $\mu_{(\tilde{K})} > s$. Thus $z_{ik} = 0$ for all $k > \tilde{K}$, and we need only consider updating those features $k \le \tilde{K}$. Notice that $\tilde{K}$ serves as a truncation level insofar as it limits the computational costs to a finite amount without approximation.

Let $K^\dagger$ be an index such that all active features have index $k < K^\dagger$ (note that $K^\dagger$ itself would be an empty feature). The computational representation for the slice sampler consists of the slice variables and the first $K^\dagger$ features: $\{s, \tilde{K}, K^\dagger, Z_{1:N,1:K^\dagger}, \mu_{(1:K^\dagger)}, \theta_{1:K^\dagger}\}$. The slice sampler proceeds by updating all variables in turn.

The slice variable is drawn from (4.46). If the value of s is less than the last represented feature presence probability $\mu_{(K^\dagger)}$, then we need to extend the representation. We sample more $\mu_{(k)}$s until $\tilde{K} < K^\dagger$ and extend Z by adding the necessary number of columns of zero, see Figure 4.6 for a pictorial explanation. In the appendix we show that the stick lengths $\mu_{(k)}$ for new features $k > K^\dagger$ can be drawn iteratively from the following distribution:

$$\begin{aligned} p(\mu_{(k)} \,|\, \mu_{(k-1)}, Z_{:,\cdot>k=0}) \propto\ & \mu_{(k)}^{\alpha-1}(1-\mu_{(k)})^N \exp(\alpha \textstyle\sum_{i=1}^N \frac{1}{i}(1-\mu_{(k)})^i) \\ & \mathbb{I}\{0 \leq \mu_{(k)} \leq \mu_{(k-1)}\}, \end{aligned} \tag{4.51}$$

which is obtained by conditioning on the fact that there are no non-zero entries beyond the kth column, eq. (4.29). We can use ARS to draw samples from (4.51) since it is log-concave in $\log \mu_{(k)}$. Finally, parameters for the new represented features are drawn from the prior $\theta_k \sim H$.

Given s, we only need to update z_{ik} for each $i = 1, \ldots, N$ and $k = 1, \ldots, \tilde{K}$. The conditional probabilities are:

$$p(z_{ik} = 1 \,|\, s, \mu_{(k)}, Z_{-ik}, \theta_{1:K^\dagger}) \propto \frac{\mu_{(k)}}{\mu^*} f(\mathbf{x}_i \,|\, z_{ik} = 1, Z_{-ik}, \theta_{1:K^\dagger}) \tag{4.52}$$

The μ^* denominator is needed when different values of z_{ik} induces different values of μ^* (e.g. if $z_{ik} = 1$ prior to the update, and was the only non-zero entry in the last non-zero column of Z).

For $k = 1, \ldots, K^\dagger - 1$, combining (4.42) and (4.43), the conditional probability of $\mu_{(k)}$ is

$$\begin{aligned} p(\mu_{(k)} \,|\, \mu_{(k-1)}, Z) \propto & \mu_{(k)}^{m_{\cdot,k}-1}(1-\mu_{(k)})^{N-m_{\cdot,k}} \\ & \mathbb{I}\{\mu_{(k+1)} \leq \mu_{(k)} \leq \mu_{(k-1)}\} \end{aligned} \tag{4.53}$$

where $m_{\cdot,k} = \sum_{i=1}^N z_{ik}$. For $k = K^\dagger$, in addition to taking into account the probability of $z_{:,K^\dagger} = 0$, we also have to take into account the probability that all columns of Z beyond $K^\dagger$ is zero. The resulting conditional probability of $\mu_{(K^\dagger)}$ is given by (4.51) with $k = K^\dagger$. Drawing from both (4.53) and (4.51) can be done using ARS. The conditional probability of θ_k for each $k = 1, \ldots, K^\dagger$, is as given before, eq. (4.35).

The feature columns are sorted in a specific way in the stick-breaking representation, therefore they are no longer exchangeable. This can result in poor mixing over the features, similar to the DP case as pointed out by Porteus et al. (2006). This problem becomes more apparent when one of the features with a large index happens to contribute highly to the likelihood. In this case, this feature would persist, which would require keeping all other unnecessary features in between the representative ones even though they are not active. We can consider moves that facilitate mixing over the feature indices and use Metropolis-Hastings sampling to evaluate the proposals similar to the ones proposed for the DP case by Porteus et al. (2006). However, the additional constraint of the stick lengths being in a strictly decreasing order complicates the formulation of the moves. In the following section, we describe an alternative method to facilitate mixing in the stick-breaking representation.

Algorithm 15 Slice sampling for stick-breaking IBP

```
The state of the Markov chain consists of the infinite feature matrix $Z$, the feature
presence probabilities $\mu_{(1:\infty)} = \mu_{(1)}, \mu_{(2)}, \ldots$ corresponding to each feature column
and the set of infinitely many parameters $\Theta = \theta_{1:\infty}$.
Only the $K^\dagger$ columns of $Z$ up to and including the last active column and the corresponding parameters are represented.

Repeatedly sample:
for all rows $i = 1, \ldots, N$ do {Feature updates}
  Sample a slice $s$ uniformly between 0 and the stick length of the last active component.
  if $s < \mu_{(K^\dagger)}$ then
    Extend the representation by breaking the stick until $s > \mu_{(K^\dagger)}$ using eq. (4.51)
    Sample parameters for the new represented features from the prior
  end if
  for all columns $k = 1, \ldots, K^\dagger$ do {update features above the slice}
    if $\mu_{(k)} > s$, update $z_{ik}$ using the full conditional eq. (4.52)
  end for
end for
for all columns $k = 1, \ldots, K^\dagger$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35)
end for
for all columns $k = 1, \ldots, K^\dagger - 1$ do {Update feature presence probabilities}
  Update $\mu_k$ by sampling from its conditional posterior, eq. (4.53) using ARS
end for
for column $K^\dagger$, the last represented column, update $\mu_{(K^\dagger)}$, by sampling from its
conditional posterior, eq. (4.51)
```

4.2.6 Change of Representations and Slice Sampling for the Semi-Ordered Stick-Breaking

Both the stick-breaking construction and the standard IBP representation are different representations of the same nonparametric object. Therefore, it is possible to make use of both representations in calculations by changing from one representation to the other (Teh, Görür, and Ghahramani, 2007). More precisely, given a posterior sample in the stick-breaking representation it is possible to construct a posterior sample in the IBP representation and vice versa.

We use the infinite limit formulation of both representations to derive the appropriate procedures. Note that for IBP, the feature labels in an arbitrarily large finite model are ignored. On the other hand, the stick-breaking construction is obtained by explicitly representing the feature presence probabilities and enforcing an ordering of the feature indices with decreasing stick lengths. Therefore, for changing from the stick-breaking to the IBP representation we simply drop the stick lengths as well as all the inactive features, leaving only the $K^{\ddagger}$ active feature columns along with the corresponding parameters. To change from IBP back to the stick-breaking representation, we have to draw the stick lengths and sort the features in decreasing stick lengths, introducing empty features if required.

We consider the finite binary matrix of Section 4.1.1 with the same distributional assumptions. We index the $K^{\ddagger}$ active features with $k = 1, \ldots, K^{\ddagger}$. Let $Z_{1:K^{\ddagger}}$ be the feature presence matrix, that is, the matrix composed of only the active feature columns. Suppose that we have $K \gg K^{\ddagger}$ features in the finite model. For the active features, the posterior for the feature presence probabilities are simply

$$\mu_k^+ \mid z_{:,k} \sim \mathrm{Beta}(\frac{\alpha}{K} + m_{\cdot,k}, 1 + N - m_{\cdot,k}), \tag{4.54}$$

see eq. (4.5). Taking the limit as $K \to \infty$, the posterior becomes,

$$\mathrm{Beta}(m_{\cdot,k}, 1 + N - m_{\cdot,k}). \tag{4.55}$$

In the stick breaking representation, we need to represent at least all features up to and including the last active feature. Therefore, the representation may include some inactive features. When changing the representation from IBP to the stick-breaking, it is sufficient to represent only those empty features with stick lengths larger than $\min_k \mu_k^+$. Thus we consider a decreasing ordering $\mu_{(1)}^{\circ} > \mu_{(2)}^{\circ} > \cdots$ on the stick lengths of the inactive components. For each $\mu_{(k)}^{\circ}$, we condition on the fact that there are no active features beyond that feature. Thus, considering infinitely many features, the density for $\mu_{(k)}^{\circ}$ is given by (4.51). ARS can be used to draw $\mu_{(1:K^{\circ})}^{\circ}$ until $\mu_{(K^{\circ})}^{\circ} < \min_k \mu_k^+$. We sample parameters for the newly represented inactive features from the prior. The stick-breaking representation is obtained by reordering $\mu_{1:K^{\ddagger}}^+, \mu_{(1:K^{\circ})}^{\circ}$ in decreasing order, with the feature columns and parameters taking on the same ordering (columns and parameters corresponding to empty features are set to 0 and drawn from their prior respectively), resulting in $K^{\dagger} = K^{\ddagger} + K^{\circ}$ features in the stick-breaking representation. The validity of this representation can be seen by referring to the connection of the IBP to the beta process Thibaux and Jordan (2007).

Semi-Ordered Stick-Breaking

In deriving the change of representations from the IBP to the stick-breaking representation, we made use of an intermediate representation whereby the active features are unordered, while the empty ones have an ordering of decreasing stick lengths. It is in fact possible to directly work with this representation, which we shall call semi-ordered stick-breaking.

The Z matrix consists of $K^{\ddagger}$ active and unordered features, as well as an ordered sequence of infinitely many empty features. The stick lengths for the active features have conditional distributions:

$$\mu_k^+ \mid z_{:,k} \sim \mathrm{Beta}(m_{\cdot,k}, 1 + N - m_{\cdot,k}) \tag{4.56}$$

while for the empty features we have a Markov property:

$$p(\mu^\circ_{(k)} \,|\, \mu^\circ_{(k-1)}) \propto (\mu^\circ_{(k)})^{\alpha-1}(1-\mu^\circ_{(k)})^N \\ \exp(\textstyle\sum_{i=1}^N \frac{1}{i}(1-\mu^\circ_{(k)})^i))\mathbb{I}\{0 \leq \mu^\circ_{(k)} \leq \mu^\circ_{(k-1)}\}. \quad (4.57)$$

Note that this equation is the same as eq. (4.51), with the conditioning on the rest of Z being inactive is inherent in the definition of μ°.

Slice Sampler

To use the semi-ordered stick-breaking construction as a representation for inference, we can again use the slice sampler to adaptively truncate the representation for empty features. This gives an inference scheme which works in the non-conjugate case, is not approximate, has an adaptive truncation level, but without the restrictive ordering constraint of the stick-breaking construction, summarized in Algorithm 16.

The representation consists only of the active features and the features and stick lengths associated with these features. The slice variable is defined as

$$s \sim \text{Uniform}[0, \mu^*] \qquad\qquad \mu^* = \min\left\{1, \min_{1\leq k\leq K^\ddagger} \mu^+_k\right\} \quad (4.58)$$

Once a slice value is drawn, we extend the representation by generating K° empty features, with their stick lengths drawn from (4.57) until $\mu^\circ_{(K^\circ+1)} < s$. The associated feature columns $Z^\circ_{K^\circ}$ are initialized to 0 and the parameters $\theta^\circ_{1:K^\circ}$ are drawn from their prior. Sampling for the matrix entries and the parameters proceed as before. Afterwards, we remove the zero columns and the corresponding parameters and stick lengths from the representation. Finally, the stick lengths for the new list of active features are drawn from their conditionals (4.56).

We have presented several MCMC algorithms for inference on the IBLF models. The important question is which of these algorithms to use in practice. In the next section, we give an empirical comparison of some of the samplers.

4.3 Comparing Performances of the Samplers

We have described several sampling algorithms for inference on the models using IBP in the previous section. It is important to have an intuition about the comparative performance of the different samplers when choosing which one to use in practice. An especially interesting question is how the computational cost is effected when non-conjugate samplers are used. Therefore, we compare the mixing performance of the conjugate Gibbs sampler (described in Algorithm 11) to the performance of the slice sampler using the strictly decreasing ordering of the stick lengths (Algorithm 15) and using the semi-ordered stick-breaking representation (Algorithm 16). We also compare the results of the the approximate Gibbs sampler (Algorithm 12) to the conjugate Gibbs sampler

Algorithm 16 Slice sampling for the semi-ordered IBP

```
The state of the Markov chain consists of the infinite feature matrix $Z$, the feature presence probabilities $\mu_{1:\infty} = \mu_1, \ldots, \mu_\infty$ corresponding to each feature column and the set of infinitely many parameters $\Theta = \{\theta_k\}_1^\infty$.
Only the $K^\ddagger$ active columns of $Z$ up to and including the last active column and the corresponding parameters are represented.

Repeatedly sample:
Change to SB representation:
Sample $\mu$s for active features ($\mu^+$) from their posterior, eq. (4.56)
Sample $\mu$s for inactive components ($\mu^\circ$) using eq. (4.57) until the smallest $\mu^+$ is larger than the smallest $\mu^\circ$
Sort columns to have $\mu$s in decreasing order
for all $i = 1, \ldots, N$ do
  Do feature updates in the SB representation
end for
Change to IBP representation:
Remove feature presence probabilities from the representation
Remove inactive feature columns from the representation
for all columns $k = 1, \ldots, K^\dagger$ do {Parameter updates}
  Update $\theta_k$ by sampling from its conditional posterior, eq. (4.35)
end for
```

results to have a sense of the accuracy of the approximation. Since for both cases the conjugate Gibbs sampling is taken as a basis of comparison, we use a conjugate model.

We choose to use the linear-Gaussian binary latent feature model from Griffiths and Ghahramani (2005). The model is summarized below, see the referred paper for a detailed description. Each data point $\mathbf{x}_i$ is assumed to be generated by a combination of a subset of the rows of A and distorted by spherical Gaussian noise,

$$\mathbf{x}_i = \mathbf{z}_i A + \boldsymbol{\varepsilon}, \tag{4.59}$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma_x^2 \mathrm{I})$. The infinite dimensional binary vector $\mathbf{z}_i$ encodes which features are contributing to $\mathbf{x}_i$, and A is a matrix (with infinitely many rows) whose kth row corresponds to the parameters for the kth feature. This model can be interpreted as a binary factor analyzer with an infinite dimensional latent space. The distribution over the whole data matrix X can be written as a matrix Gaussian,

$$X \mid Z, A, \sigma_x \sim \mathcal{N}(ZA, \, \sigma_x^2 \mathrm{I}). \tag{4.60}$$

Entries of A are drawn i.i.d. from a zero-mean Gaussian with variance σ_A^2. We can denote the distribution over a finite $(K \times D)$ part of A also as matrix Gaussian,

$$A \mid \sigma_A \sim \mathcal{N}(\mathbf{0}, \sigma_A^2 \mathrm{I}), \tag{4.61}$$

where $\mathbf{0}$ denotes a $K \times D$ matrix of zeros. We put an IBP(α) prior on the latent binary feature matrix Z, and a gamma prior on the IBP parameter $\alpha \sim \mathcal{G}(1, 1)$, completing the model.

We compare the mixing performance of the two slice samplers (on the strictly decreasing weights, and on the semi-ordered weights) and Gibbs sampling described in the previous section. We chose to apply the samplers to simple synthetic data sets so that we can be assured of convergence to the true posterior and that mixing times can be estimated reliably in a reasonable amount of computation time for all samplers. We also chose to use a conjugate model since Gibbs sampling requires conjugacy. However, note that our implementation of the two slice samplers did not make use of the conjugacy.

We generated $1, 2$ and 3 dimensional data sets from the model with data variance fixed at $\sigma_x^2 = 1$, varying values of the strength parameter $\alpha = 1, 2$ and the latent feature variance $\sigma_A^2 = 1, 2, 4, 8$. For each combination of parameters we produced five data sets with 100 data points, in total 120 data sets. For all data sets, we fixed σ_x^2 and σ_A^2 to the generating values and learned the feature matrix Z and α.

We are interested in how the samplers on the nonparametric part of the model perform Therefore, we fix the σ_x and σ_A values and learn Z, A and α for all cases. For each data set and each sampler, we repeated 5 runs of $15,000$ iterations. We used the autocorrelation coefficients of the number of represented features $K^\ddagger$ and α (with a maximum lag of 2500) as measures of mixing time. We found that mixing in $K^\ddagger$ is slower than mixing in α for all data sets and all three samplers. We also found that in this regime the autocorrelation times do not vary with dimensionality or with σ_A^2. In Figure 4.7 we show the autocorrelation times of α and $K^\ddagger$ over all runs, all data sets, and all three samplers. As expected, the slice sampler using the decreasing stick lengths ordering was always slower than the semi-ordered one. Surprisingly, we found that the semi-ordered slice sampler was just as fast as the Gibbs sampler which fully exploits conjugacy. This is about as well as we would expect a more generally applicable non-conjugate sampler to perform. This is a motivating result for using the semi-ordered slice sampler for inference on complex non-conjugate IBLF models.

The first algorithm introduced for inference on the non-conjugate IBLF models is the approximate Gibbs sampling algorithm (Algorithm 12). Even though efficient sampling techniques that do not need to use an approximation have been developed, it is interesting to have an insight about how the approximate method performs compared to the non-approximate ones. We compare the modeling performance of Algorithm 12 to the conjugate Gibbs sampling results using the model described above.

We generated a synthetic data set of 6×6 images described in Griffiths and Ghahramani (2005). The input images are composed of a combination of (a subset of) four latent features and zero-mean Gaussian noise with 0.5 standard deviation. We used the Gibbs sampling for conjugate models and the approximate Gibbs sampling for non-conjugate models to learn the latent structure. For the approximate scheme we used

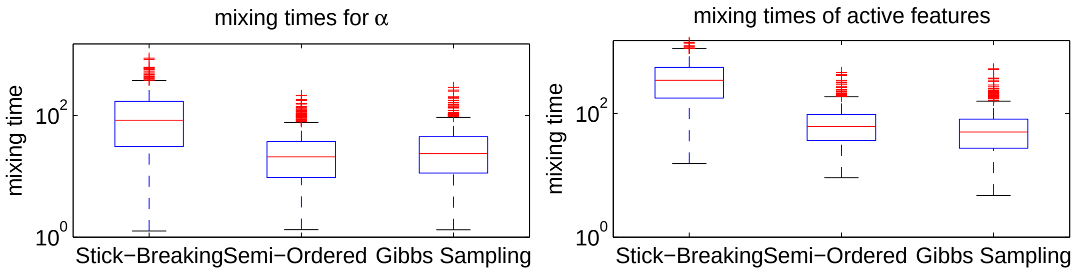

Figure 4.7: Comparison of the mixing times for α (left) and the number of active features $K^\ddagger$ (right) for the slice sampler on stick-breaking representation with decreasing stick lengths, the slice sampling for the semi-ordered stick-breaking representation, and for the conjugate Gibbs sampler. As expected, mixing is slowest when the stick lengths are constrained to have the strictly decreasing ordering. The slice sampler on the semi-ordered stick lengths not exploiting conjugacy is as fast as the Gibbs sampler using conjugacy.

five auxiliary features for unique feature updates. Since the approximate scheme is more sensitive to hyperparameter values, we fixed the values of α and σ_A for the initial 25 iterations for both samplers. The mixing time of the approximate scheme is slower as expected, since only a finite number of new features with specific parameters are considered instead of integrating over the infinitely many features. Nevertheless, the modeling performance of the approximate scheme is found to be comparable to the exact scheme. The latent features making up the images and the generating noise scale can be accurately recovered. The trace plots of both samplers are depicted in Figure 4.8 and Figure 4.9.

4.4 A Flexible Infinite Latent Feature Model

The binary feature model used in the previous section models the observations as a combination of the latent features. Each latent feature is either present with weight 1 or does not contribute at all to the observation. We can obtain a more powerful model by allowing the latent features to take on real values while preserving sparsity. We use a sparse real-valued latent feature matrix F composed of two components: a binary matrix Z that encodes the presence or absence of the features, and a real-valued matrix Y expressing the weight of each feature for each object, as suggested in Griffiths and Ghahramani (2005). In particular, we define Y to be a matrix of the same size as Z, with i.i.d. zero mean unit variance Gaussian entries. We model each $\mathbf{x}_i$ as

$$(\mathbf{x}_i|\mathbf{z}_i, \mathbf{y}_i, A, \sigma_x^2) \sim \mathcal{N}((\mathbf{z}_{i,:} \otimes \mathbf{y}_{i,:})A, \sigma_x^2\mathrm{I}), \tag{4.62}$$

where $\otimes$ denotes elementwise multiplication. Specification for the rest of the parameters A, σ_x, σ_A and α is as in the previous section.

Since the data distribution eq. (4.62) does not depend on the zero columns of Z, the

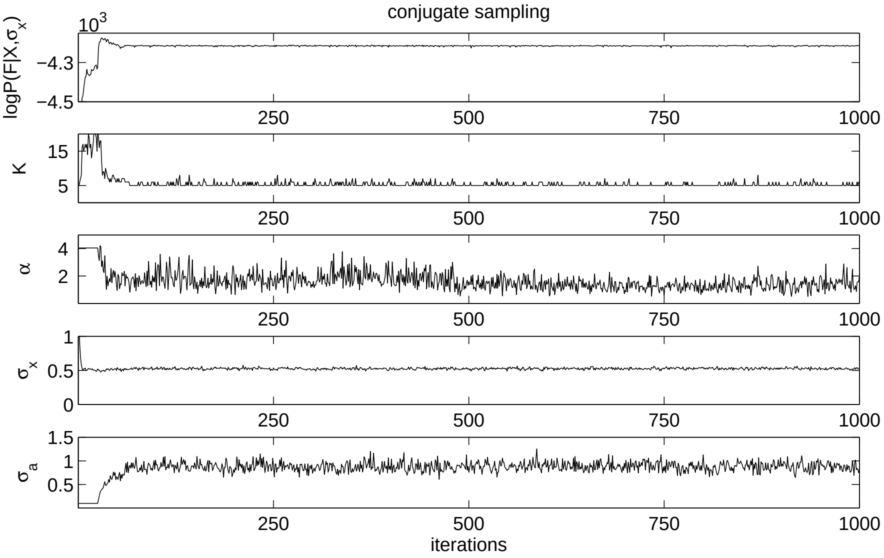

Figure 4.8: Trace plots for the Gibbs sampling using conjugacy. The sampler converges in a few iterations to the high probability regions, employing five to seven latent features. The generating value of σ_x is recovered.

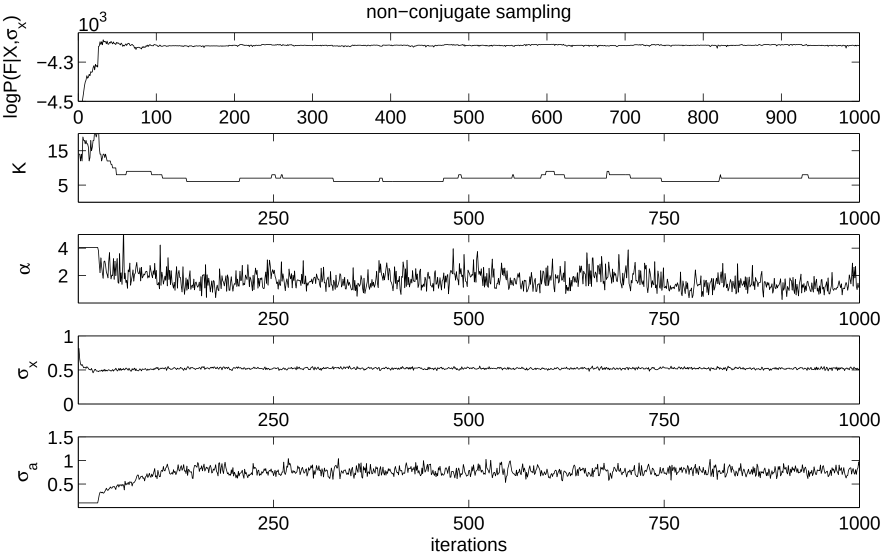

Figure 4.9: Trace plots for the approximate Gibbs sampler not exploiting conjugacy with five auxiliary features for unique feature updates. Compared to the conjugate Gibbs sampler, the chain moves slower, especially for K. However the samples for all parameters have similar values to the samples from the conjugate Gibbs sampler.

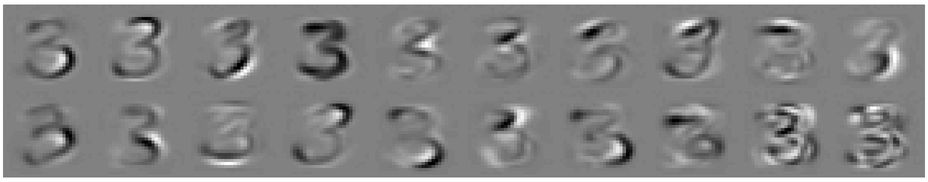

Figure 4.10: Features that are shared between many images. Observe that different segments are pronounced in different features.

effective latent dimension is determined by the number of nonzero columns of Z, similar to the simpler model in the previous section. This model has two sets of parameters, Y and A, associated with Z. We can not integrate over both of them. Therefore we need an algorithm for non-conjugate IBP models for inference on this model. Motivated by the results of the previous section, we choose to apply the semi-ordered slice sampler to learn the latent features.

We modeled 1000 examples of the digit 3 in the MNIST data set. The digit images are first preprocessed by projecting on to the first 64 PCA components, and the sampler is ran for 10000 iterations. We show results for the digit 3. The distribution of the number of nonzero features and the trace plots of the log likelihood are shown in Figure 4.12.

The model succesfully finds latent features to reconstruct the images as shown in Figure 4.11. Some of the latent features found are shown in Figure 4.10. These appear to model edge segments of the digit 3.

4.5 A Choice Model with Infinitely Many Latent Features

Modeling choice is an important topic of study for psychology, economics and related sciences. The goal is to understand and model the behavioral process that leads to the subject's choice. In a choice scenario, the subject is presented with more than one alternative from a *choice set* and is asked to choose one of the alternatives. The *alternatives* may be items or courses of actions.

The answer to the question *how much* can be treated using regression models. The choice models address the question *which*. Ordinal regression can be seen as a choice model for which the alternatives have a particular ordering. Here, we will consider discrete choice models where there is no particular ordering of the alternatives. Discrete choice models assume there to be finitely many alternatives, all of which are included in the choice set. The alternatives are assumed to be mutually exclusive, that is, choosing one alternative implies not choosing any other. As data, we have outcomes of repeated choice from subsets of the choice set. We use the number of times each alternative is chosen over some others. We want to learn the true choice probabilities.

Psychologists have long been interested in the mechanisms underlying choice behavior (Luce, 1959). In virtually all psychological experiments subjects are (repeatedly) asked to make a choice and the responses are recorded. Often the choice is very simple like

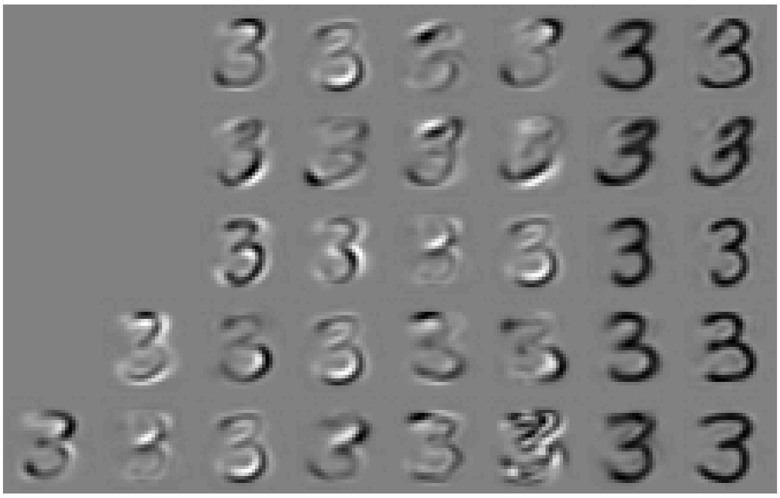

Figure 4.11: Examples of reconstructed images and their latent features. Last column: original digits, second last column: reconstructed digits, other columns: features used for reconstruction. For each image, the binary vector $\mathbf{z}_i$ determines the presence or absence of a feature. The real valued vector $\mathbf{z}_i$ determines how much each feature contributes to the observed image.

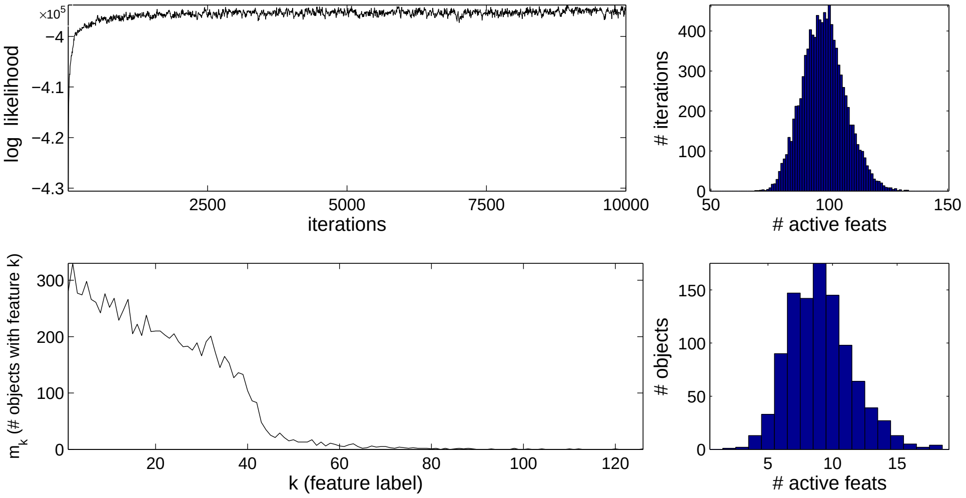

Figure 4.12: Distribution of the number of active features over 10000 iterations (left). Distribution of number of active features for each input at one iteration (right). Note that the number of features that a particular data point has is much less than the total number of active features. The number of observations possessing each feature at one iteration. Note that although the total number of active features is much larger, about half of the features belong to only a few observations.

pressing one of two buttons. However, even in these simple choices one finds probabilistic responses. In what seem to be identical experimental conditions, one can observe that different subjects respond differently. Responses vary even for the same subject that is repeatedly presented with the same choice scenario. Choice is an omnipresent phenomenon in economics as well. Consumers choose one brand instead of another, commuters choose to take the bus rather than the car and college students prefer one university over another. Considerable effort has been put into probabilistic modeling of data arising from such choice situations (McFadden, 2000; Train, 2003).

In accordance with economic theory, choice models in economics often assume that humans are rational and make choices by maximizing utility. The reason for the observed probabilistic variations in choice behavior is the random variations in utility. These variations may arise because different decision makers make different judgments about the utility of an option, or because each decision maker varies randomly in her assessment of utility over time. Models that fit into this framework are called random utility models (RUMs). In contrast to RUMs many psychological models do not assume a rational decision maker—instead they attempt to explain the (probabilistic) mental processes that take place in the course of a decision.

In spite of the differences in the approaches, the models proposed in these two different fields have correspondences. One of the most influential psychological models is the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959), which can be interpreted also as the random utility model logit (Tversky, 1972). However, it is well-known that the BTL (or logit) cannot account for all of the choice patterns that can be observed in choice outcomes. There is a large literature on how choice data can violate the assumptions built into the BTL model (Restle, 1961; Rumelhart and Greeno, 1971; Tversky, 1972; Train, 2003).

Within the RUM framework several other models have been suggested to cope with more general choice patterns, e.g. the probit model with correlated noise, the nested logit and mixed logit models (Train, 2003). Recently, a mixed multinomial logit model with a non-parametric mixing distribution has been proposed by James and Lau (2004) using the Dirichlet process.

Similarly, there have been more general models developed in psychology. The elimination by aspects (EBA) model (Tversky, 1972) —which includes the BTL model as a special case—is a well known choice model in psychology. The correspondences of the EBA model and some RUMs are discussed in Batley and Daly (2006). We focus on the EBA model in this section.

The EBA model assumes the options to be represented by binary feature vectors, called aspects. If the choice was between several mobile phones the features could be whether they have a built-in MP3-player, the display is in color, the battery lasts long enough, etc. Each feature has a weight associated with it reflecting the importance of each aspect for the choice. The subject selects a feature at random (but more important features have a higher probability of being selected) and eliminates all options that do not have this feature. This process is repeated until only one option remains. If the features are known their weights can be estimated from choice data (Wickelmaier and Schmid, 2004). However, generally the features that the subject considered are not available to the researcher and one would like to infer them from observed choices. Although the EBA model can account for many choice scenarios, its usefulness for the analysis of choice data has been extremely limited by the fact that inference about the underlying features is very difficult (Tversky and Sattath, 1979; Batsell et al., 2003).

In the following, we present a Bayesian formulation and analysis of the EBA model. We employ the IBP as a prior over the latent binary features of alternatives in the choice set and infer the latent features and their weights (importance of each feature to the choice maker) using MCMC and demonstrate the performance of the proposed model on a data set studied in psychology. We describe the non-parametric Bayesian formulation of the EBA model in the next section. The MCMC algorithm used for inference on the model is presented in Section 4.5.2, followed by experimental results in Section 4.5.3, and we conclude with a discussion in Section 4.5.4.

4.5.1 Model Specification

The EBA model is defined for choice from a set of several options but for clarity of presentation we consider only the paired comparison case here which reduces the EBA model to Restle's choice model (Restle, 1961; Tversky, 1972). Nevertheless, the inference techniques we describe below are also valid for the general EBA model.

In a paired comparison experiment there is a set of N options but the subject is presented only with pairs of options at a time. The task of the subject is to indicate which of the two options i and j she prefers. In the EBA model, options are described by K-dimensional binary vector of features $\mathbf{z}$, referred to as aspects. The probability of choosing option i over option j is given as

$$p_{ij} = \frac{\sum_k w_k z_{ik}(1 - z_{jk})}{\sum_k w_k z_{ik}(1 - z_{jk}) + \sum_k w_k z_{jk}(1 - z_{ik})}, \tag{4.63}$$

where z_{ik} denotes the binary value of the kth feature of option i and w_k is the positive weight associated with the kth feature. The greater the weight of a feature, the heavier its influence on the choice probabilities. The sum $\sum_k w_k z_{ik}(1-z_{jk})$ collects the weights for all the aspects that option i has but option j does not have. Therefore, the choice between two alternatives depends only on the features that are not shared between the two. Equation (4.63) expresses that the EBA model can account for the effects of similarity on choice. For the above equation, we define the ratio $\frac{0}{0} = 0.5$. That is, if two alternatives have exactly the same set of features, the choice probability is 0.5. On the other extreme, if the options are characterized only by unique aspects, i.e. no option shares any feature with any other option, the BTL model is recovered. Recently, Ruan et al. (2005) have developed a choice model using a Poisson race model in which the choice probabilities are mainly determined by the difference in the features of the alternatives, but the similarities are also allowed to have some effect. This model can be seen as a generalization of the EBA model.

If one option i has all the features that another alternative j has and more features on top of these then i will always be preferred over j. This is a reasonable assumption but in real choice data it can happen that subjects occasionally fail to choose alternative

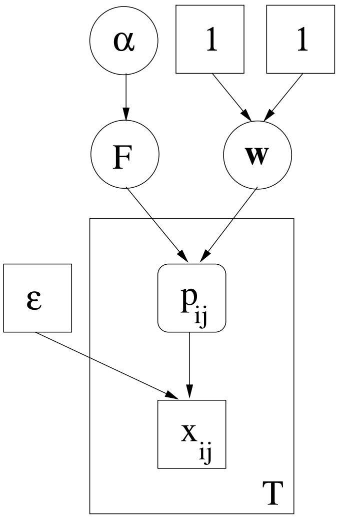

Figure 4.13: Graphical representation of the elimination by aspects model with IBP prior.

i because of error or lack of concentration (Kuss et al., 2005). In order to make our inference more robust to these lapses we add a lapse probability ε to the choice probabilities that we assume to have a small value. Thus the choice probabilities become $\tilde{p}_{ij} = (1-\varepsilon)p_{ij} + 0.5\varepsilon$.

Let us denote the number of times that i was chosen over j in a paired comparison experiment by x_{ij}. It is assumed that x_{ij} is binomially distributed

$$P(x_{ij} \,|\, \mathbf{z}_i, \mathbf{z}_j, \mathbf{w}) = \binom{x_{ij}+x_{ji}}{x_{ij}} (\tilde{p}_{ij})^{x_{ij}} (1-\tilde{p}_{ij})^{x_{ji}}, \tag{4.64}$$

and is independent of all other comparisons in the experiment. We can therefore write the likelihood of all the observed choices in a paired comparison experiment as

$$P(X|Z,\mathbf{w}) = \prod_{j=1}^{N} \prod_{i<j} P(x_{ij}|\mathbf{z}_i, \mathbf{z}_j, \mathbf{w}), \tag{4.65}$$

where X is a matrix that collects the results of all paired comparisons x_{ij}, Z is a $N \times K$ binary matrix of features with entries z_{ik} for the kth feature of option i, and $\mathbf{w}$ is a vector containing the weights w_k of all features.

To complete the model we need to specify the priors over the binary features and the weights. Since we do not know the features a priori we use an IBP prior over the binary feature matrix and independent gamma priors on the weights,

$$Z \sim \mathrm{IBP}(\alpha) \tag{4.66}$$

$$w_k \sim \mathcal{G}(1,1). \tag{4.67}$$

See Figure 4.13 for a graphical representation of the model.

4.5.2 Inference using MCMC

Inference for the above model can be done using MCMC techniques. Görür et al. (2006) present results using the approximate Gibbs sampling for updating the feature matrix Z. Here, we use slice sampling with the semi-ordered stick-breaking representation. We use Gibbs sampling for the IBP parameter α and Metropolis Hastings updates for the weights $\mathbf{w}$.

Gibbs sampling for the feature updates requires the posterior of each z_{ik} conditioned on all other features $Z_{-(ik)}$ and the weights $\mathbf{w}$. The conditional posterior for the entries of the feature matrix can be obtained by combining the likelihood given in eq. (4.65) with the prior feature presence probability $\mu_{(k)}$,

$$P(z_{ik} = 1 \,|\, Z_{-ik}, \mathbf{w}, \mu_{(k)}) \propto \mu_{(k)} P(X \,|\, z_{ik} = 1, Z_{-ik}, \mathbf{w}, \mu_{(k)}). \tag{4.68}$$

We update the weights using Metropolis Hastings sampling. We sample a new weight from a proposal distribution $Q(w'_k|w_k)$ and accept the new weight with probability

$$\min\left(1, \frac{P\big(w'_k|X,\, Z,\, \mathbf{w}_{-k},\, w_k,\, \lambda\big)}{P\big(w_k|X,\, Z,\, \mathbf{w}_{-k},\, w'_k,\, \lambda\big)} \frac{Q(w_k|w'_k)}{Q(w'_k|w_k)}\right). \tag{4.69}$$

As the proposal distribution we use a gamma distribution with mean equal to the current value of the weight, w_k, and standard deviation proportional to it,

$$Q(w'_k|w_k) = \mathcal{G}(\eta w_k, \eta / w_k). \tag{4.70}$$

We adjust η to have an acceptance rate around 0.5 initially.

Note that there are infinitely many weights that are associated with the infinitely many features. Since the inactive features and their weights do not affect the likelihood, we need to only represent and update the weights that are associated with the active features.

4.5.3 Experiments

In this section, we present empirical results on an artificial and a real data set. Both data sets have been considered in the choice model literature.

We initialize the parameters α, Z and $\mathbf{w}$ randomly from their priors and set the lapse parameter to $\varepsilon = 0.01$.

Paris-Rome

We first consider a synthetic example given by Tversky (1972). It was constructed as a simple example that the BTL model cannot deal with. We will use this example to illustrate that the EBA model with infinitely many latent features can recover the latent structure from choice data.

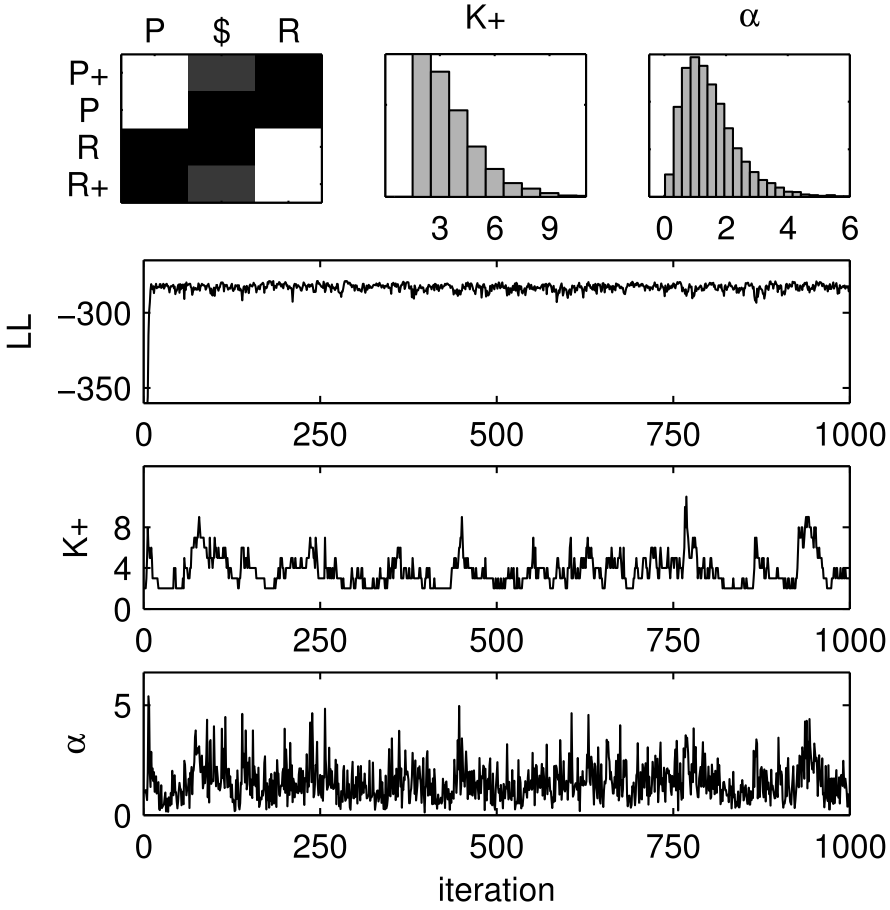

Figure 4.14: Feature matrix representation and simulation results on the toy data: the choice between trips to Paris and Rome. Top left: Features weighted by the associated values are shown. Rows correspond to the alternatives and columns correspond to the features. Darker means smaller in amplitude. The alternatives P and $P+$ share the feature of being the trip to Paris and R and $R+$ share the feature of being the trip to Rome. $P+$ and $R+$ share the small bonus denoted by the $ column.

Table 4.1: Choice probabilities for the Paris-Rome example. Columns are chosen over rows.

	P+	P	R	R+
P+	0.50	0	0.48	0.50
P	1	0.50	0.50	0.52
R	0.52	0.50	0.50	1
R+	0.50	0.48	0	0.50

Consider the choice between two trips that are on offer at a travel agency: one to Paris (P) and one to Rome (R). Another travel agency offers exactly the same trips to Paris and to Rome except that there is an additional small bonus. We denote these options by $P+$ and $R+$, respectively. The options in the choice set consist of P, $P+$, R, $R+$. We assume that the decision maker assigns the same value to the trip to Paris and to the trip to Rome. Hence, she will be equally likely to prefer the trip to either city. We denote the probability that Paris is chosen over Rome with $P(P, R) = 0.5$. As it is always better to get a bonus than to not get it we can assume that when given the options P and $P+$ she would choose $P+$ with certainty, i.e. $P(P+, P) = 1$. However, since a small bonus will not influence the choice between the two cities, $P(P+, R)$ will be close to 0.5, and likewise for $P(R+, P)$. We assume that the trip itself to either city has a feature with value w and the bonus is worth $0.01w$. Thus, the binary feature matrix weighted with the importance of the features can be represented as shown in the top left of Figure 4.14. The alternatives P and $P+$ share the feature of being the trip to Paris and R and $R+$ share the feature of being the trip to Rome. $P+$ and $R+$ share the small bonus denoted by the \$ column. Note that there might be some features that trips to either city possess such as taking the plane, going to Europe, etc. Since these features will be common to all options in the choice set they do not affect the choice probabilities. The matrix of the deterministic choice probabilities calculated assuming the EBA model using eq. (4.63) is shown in Table 4.1.

We generated choice data of 100 comparisons for each pair using these probabilities and used the EBA model with infinitely many latent features (iEBA) to infer the latent structure of the options. The results of a sample run for the iEBA model are shown in Figure 4.14. The top row shows the feature matrix that is used to generate the data, histogram of the frequencies for the active features and the posterior distribution of the IBP parameter α. The plots below show the change in the log likelihood, the number of active features of the IBP part of the feature matrix and α over the sampling iterations. It can be seen from the trace plots that the burn-in time is very short and the chain seems to mix well.

The iEBA model takes only the choice data as input, and it can recover the feature matrices given in Figure 4.14 from the data. Some sample feature matrices from the chain are depicted in Figure 4.15 which shows that the model successfully finds the latent feature representation that was used to generate the data. Note that the mapping from the choice probabilities to features is not unique, that is, several latent feature

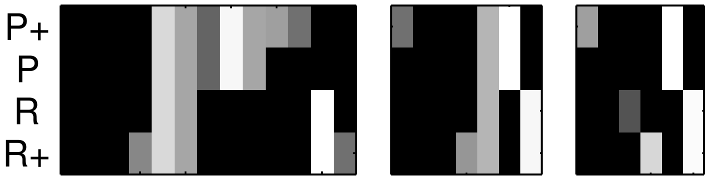

Figure 4.15: Random binary feature matrix samples weighted with the feature weights from the chain for the Paris-Rome example. The gray level indicates the weight of each feature. Rows correspond to options and columns to features. Only the active features are shown. It is inferred that there are features that only P and $P+$ share, and only R and $R+$ share. Note that there are also some features common to all options which do not affect the likelihood. The first four columns in these figures are the unique features, remaining columns are from the posterior of the infinite part. Note that the weights of the unique features for P and R are very close to zero, which is expected since they are not supported by the data. The unique features for $P+$ and $R+$ have small weights representing the small bonus.

representations may result in the same choice probabilities. The important point is that the model can learn the true choice probabilities by inferring that there are features that only P and $P+$ share, and there are features that only R and $R+$ share. The small bonus that $R+$ and $P+$ have in common is represented as two different features with similar weights. Models on a larger set of alternatives might result in feature representations that cannot be interpreted easily, as will be seen in the next example.

Celebrities

As a second example we analyze real choice data from an experiment by Rumelhart and Greeno (1971) that is known to exhibit choice patterns that the BTL model cannot capture. Subjects were presented with pairs of celebrities and were asked with whom they would prefer to spend an hour of conversation. There are nine celebrities in the choice set that were chosen from three groups: three politicians, three athletes and three movie stars. Individuals within each group are assumed to be more similar to each other and this should have an effect on the choice probabilities beyond the variation that can be captured by the BTL model. The data had been collected assuming that the choice probabilities could be fully captured by a feature matrix that has unique features for each individual plus one feature for being a politician, one feature for being an athlete and one feature for being a movie star. The assumed feature matrix is shown in Figure 4.16. This feature matrix can also be depicted as a tree therefore we refer to this model as the tree EBA model (tEBA) (Tversky and Sattath, 1979).

We modeled the choice data with different specifications for the EBA model: the model which assumes all options to have only unique features (BTL), the EBA model with fixed features of tree structure (tEBA), two finite EBA models with the number of features fixed to be 12 and 15 (EBA12 and EBA15), and the EBA model with infinitely

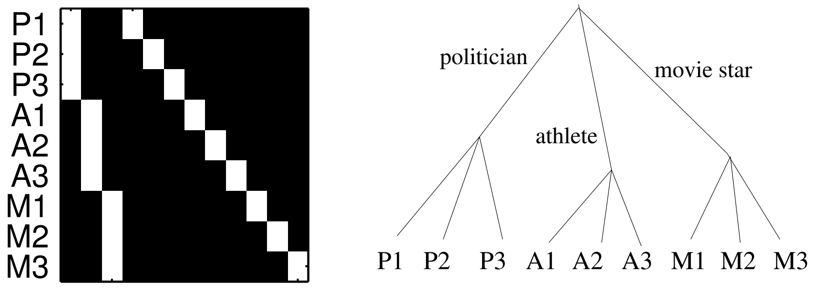

Figure 4.16: Representation of the assumed features for the celebrities data. The tree structure on the right shows nine celebrities of three professions. Each individual is assumed to have a unique feature and one feature that he shares with the other celebrities that have the same profession. The left panel shows this assumed feature matrix which is used for training the tEBA model.

many latent features (iEBA).

We compare the predictive performance of each model using a leave-one-out scenario. We train the models on the data of all possible pairwise comparisons except one pair. We then predict the choice probability for the pair that was left out and repeat this procedure for all pairs. We evaluate the performance of the models using the loss function $\mathcal{L}(\hat{\theta}, \theta) = -\theta \log \hat{\theta} - (1-\theta)\log(1-\hat{\theta})$ which expresses the discrepancy between the true probabilities θ and the predicted probabilities $\hat{\theta}$. The mean of the predicted probabilities minimizes the expected loss. Therefore we take this as a point estimate and report the negative log likelihood on the mean of the predictive distribution for each pair.

The negative log likelihood of each model averaged over the 36 pairs is shown in Table 4.2. For better comparison we also report the values for a baseline model that always predicts 0.5 and the upper bound that could be reached by predicting the empirical probabilities exactly. Furthermore, to see how much information we gain over the baseline model by using the different models we report an information score for each single paired comparison: The negative log likelihood averaged over the pairs and number of comparisons in bits with the baseline model subtracted.

The choice set was designed with the tree structure in mind. As expected, the BTL model cannot capture as much information as the other models. The iEBA and the tEBA models have the best predictive performance on average. This shows that the underlying structure could be successfully represented by the infinite model. However, we cannot observe the tree structure in the latent features that are found by the iEBA. Note that different feature representations can lead to the same choice probabilities. The mean number of active features for the iEBA model for different pairs is between 30 and 50—much more than the number of features in tEBA. This explains why the average performance of EBA12 and EBA15 is worse than that of iEBA and tEBA even though they could implement the tree structure in principle. As we cannot know how

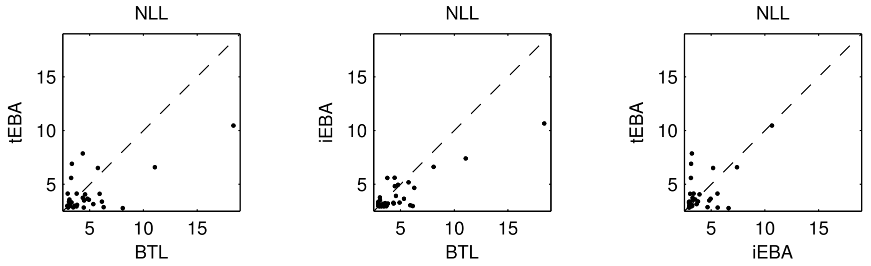

Figure 4.17: The negative log likelihood of one model versus another. Each point corresponds to one paired comparison.

Table 4.2: Predictive performance of different models: The baseline model that always predicts a probability of 0.5 (BASE), the Bradley-Terry-Luce model (BTL), the finite EBA model with 12 features (EBA12), finite EBA model with 15 features (EBA15), EBA model with tree structure (tEBA), the EBA model with IBP prior (iEBA), and for comparison the empirical probabilities (EMP). NLL: The negative log likelihood values on the mean predictive probabilities. IS: Information score in bits (the information gain compared to the model that always predicts 0.5). NLL and IS both express the same values in different scales.

MODEL	NLL	IS
BASE	17.57	0
BTL	4.66	0.0795
EBA12	4.50	0.0806
EBA15	4.31	0.0817
tEBA	3.95	0.0839
iEBA	3.92	0.0841
EMP	2.89	0.0905

many features will be necessary beforehand this is a strong argument in the favor of using a non-parametric prior.

Figure 4.17 shows a more fine-grained analysis of the BTL, tEBA and iEBA models. Each point corresponds to one paired comparison: the negative log likelihood of one model versus the negative log likelihood of another model. Out of 36 pairs BTL has a smaller log likelihood for 23 of the pairs when compared to tEBA and 26 when compared to iEBA. However, it can be seen that the bad performance of the BTL model on average is also due to the fact that it cannot capture the probabilities of some pairs at all. The iEBA and tEBA likelihoods are comparable although there are some pairs on which iEBA performs better than tEBA, and vice versa.

4.5.4 Conclusions

EBA is a choice model which has correspondences to several models in economics and psychology. The model assumes the choice probabilities to result from the non-shared features of the options. The usefulness of the EBA model has been hampered by the lack of a method that can accurately infer the unknown features of the alternatives. We have suggested to use an infinite latent feature matrix to represent the unknown features. We showed empirically that the infinite latent feature model (iEBA) can capture the latent structure in the choice data as well as the handcrafted model (tEBA). For data for which we have less prior information it might not be possible to handcraft a reasonable feature matrix.

Different feature matrices can result in the same choice probabilities and therefore are not distinguished by the model. For example, features that are shared by all options do not affect the likelihood. Furthermore, only the ratio of the weights affects the likelihood. What seems to be a non-identifiability problem is not an issue for sampling since we are interested in inferring the choice probabilities, not the "true" features.

On a more conceptual side, the non-identifiability of the features makes the samples from the posterior hard to interpret. For instance, for the celebrities data one might have hoped to find feature matrices that correspond to a tree or at least find matrices with some other directly interpretable structure. However, although the experiment by Rumelhart and Greeno (1971) was designed with the tree structure in mind Figure 4.16, we do not know the true latent features that lead to the choices of the subjects. Nevertheless, we can use the posterior to predict future choices from past data, assess the similarity of the options and cluster or rank them.

The EBA model does not take into account neither the time ordering of choice outcomes nor the identities of the choice makers. For some choice scenarios this information is irrelevant or not available. However, a model that can make use of the additional information when available would potentially be more powerful in representing the structure. This is a direction that the EBA model can be improved.

Another possible improvement of the EBA model would be to take into account the shared features as well when calculating the choice probabilities. One possibility is to add to eq. (4.63) the effect of the shared features with a scaling factor $s \in [0, 1]$. The EBA model would be recovered for $s = 0$, whereas the BTL model would be obtained when the similar features are treated the same as the differences, that is, with $s = 1$. Using a small but nonzero scale value may improve recovering the latent structure of the alternatives especially when there are pairs of alternatives that have not been compared.

4.6 Discussion

The Indian buffet process has been recently introduced by Griffiths and Ghahramani (2005), and rapidly attracted interest due to its conceptual simplicity and promising flexibility for defining nonparametric latent feature models. Its relation to the DP and related distributions has inspired models and inference algorithms from the DP literature to be adjusted to the IBP models. The different approaches with which the equivalent distribution on the binary matrix can be defined suggests generalizations of the IBP.

In this chapter, we have summarized the different approaches for defining the distribution induced by the IBP. We have described the MCMC algorithms that have been developed for inference on the IBP models. We have compared the performance of the different samplers on a simple model to have an intuition about their performances. We demonstrated the use of the IBP as a prior on two models: A sparse factor analysis model with an over-complete basis for learning latent features of handwritten digits, and a choice model with infinitely many features describing the alternatives in the choice set.

Given the performance of the proposed models, development of nonparametric models using IBP and generalizations for treating many other machine learning problems is an area for future research. There is much insight to gain from studying the connections between the IBP and related distributions, which would lead to further understanding of the theoretical and practical properties of the distribution. Development and improvement of inference algorithms on IBP models is also another direction for future research.

5 Conclusions

The analysis of real-world problems demands a principled way of summarizing the data and representing the uncertainty in these summaries. Bayesian methods allow representing prior belief about the system generating the data and provides uncertainties about the predictions. The analysis of complex systems often requires flexible models that accurately capture the dependencies in the data. Simple parametric models can become inadequate for complex real-world problems. Nonparametric methods are powerful tools that allow definition of flexible models, since they can be used to approximate any of a wide class of distributions.

The Dirichlet process (DP) is one of the most prominent random probability distributions. Although introduced in the 70s, its use have become popular only recently due to the development of sophisticated inference algorithms. The Indian buffet process (IBP) is a generalization of the related Chinese restaurant process which allows definition of more powerful models. In this thesis, we have presented empirical analysis of models using the DP and the IBP.

The DP is defined by two parameters, a base distribution and a concentration parameter. The base distribution is a probability distribution, determining the mean of the DP. The concentration parameter is a positive scalar that can be seen as the strength of belief in the prior guess. Use of conjugate priors makes computations for inference much easier for Bayesian models in general. However, conjugate priors may fail to represent one's prior belief. The trade off between the computational ease and modeling performance is an important modeling question. The Dirichlet process mixtures of Gaussians (DPMoG) model is one of the most widely used Dirichlet process mixture (DPM) models. We have empirically addressed this question in the DPMoG using conjugate and conditionally conjugate base distributions. We have compared the modeling performance and the computational cost of the inference algorithms for both prior specifications. We have empirically shown that it is possible to increase computational efficiency by exploiting conditional conjugacy while obtaining better modeling performance than the fully conjugate model.

DPM models can be seen as mixture models with infinitely many components. Although the number of components are not bounded, there is an inherent clustering property in DPM models. We formulated a nonparametric form of the mixtures of factor analyzers (MFA) model using the DP. The MFA models the data as a mixture of Gaussians with reduced parametrization. Hence, the Dirichlet process MFA (DPMFA) model allows modeling high dimensional data efficiently. We have exploited the clustering property of the DPMs for the task of spike sorting, that is, clustering of spike waveforms that arise from different neurons in an extracellular recording.

The IBP, a distribution over sparse binary matrices with infinitely many columns, has been defined as an extension of DP. An IBP can be used as a nonparametric prior over latent binary features in a hierarchical model. We have described several different approaches for defining and generalizing the distribution given by the IBP.

We have summarized the MCMC techniques developed for inference on the IBP models. The modeling performance of an IBP model is demonstrated by successfully learning the features of handwritten digits. We have formulated a nonparametric version of a choice model, elimination by aspects (EBA) and applied the slice sampling algorithm to infer latent features of alternatives in a choice experiment.

The development and assessment of sophisticated models using DP and extensions as well as inference techniques for these models is an ongoing area of research. The flexibility and statistical strength of the nonparametric Bayesian models motivates application of these models to challenging real-world problems.

A Details of Derivations for the Stick-Breaking Representation of IBP

The stick-breaking construction of the IBP has been summarized in Section 4.1.2. Here, we give details of derivation for obtaining this representation.

A.1 Densities of ordered stick lengths

The density for the kth largest feature presence probability is obtained by considering the decreasing order statistics of K i.i.d. random variables in the limit as $K \to \infty$. We define $\mu_{l_1} \geq \mu_{l_2} \geq \cdots \geq \mu_{l_K}$ to be the decreasing ordering of $\mu_1, \ldots, \mu_K$, and $\mathbf{L}_k$ to be the set of indices other than the k largest μ's,

$$\mathbf{L}_k = \{1, \ldots, K\} \backslash \{l_1, \ldots, l_k\}.$$

Thus, the indices for the k largest feature presence probabilities are $\{l_1, \ldots, l_k\}$. Denoting $\mu_{(k)} = \mu_{l_k}$, the k largest values among $\mu_{1:K}$ is denoted as $\mu_{(1:k)} = \{\mu_{(1)}, \ldots, \mu_{(k)}\}$ and we have

$$\mu_l \leq \mu_{(k)} \quad \text{for all} \quad l \in \mathbf{L}_k. \tag{A.1}$$

Thus, the range of the subsequent μ_l's given $\mu_{(k)}$ is restricted to $[0, \mu_{(k)}]$, resulting in the following pdf:

$$p(\mu_l \,|\, \mu_{(1:k)}) = \frac{\frac{\alpha}{K} \mu_l^{\frac{\alpha}{K}-1}}{\int_0^{\mu_{(k)}} \frac{\alpha}{K} t^{\frac{\alpha}{K}-1} \,\mathrm{d}t} = \frac{\alpha}{K} \mu_{(k)}^{-\frac{\alpha}{K}} \mu_l^{\frac{\alpha}{K}-1}. \tag{A.2}$$

Integrating, the cdf for μ_l conditioned on $\mu_{(1:k)}$ is found to be

$$\begin{aligned} F(\mu_l \,|\, \mu_{(1:k)}) &= \int_0^{\mu_l} \frac{\alpha}{K} \mu_{(k)}^{-\frac{\alpha}{K}} t^{\frac{\alpha}{K}-1} \mathrm{d}t \\ &= \mu_{(k)}^{-\frac{\alpha}{K}} \mu_l^{\frac{\alpha}{K}} \mathbb{I}(0 \leq \mu_l \leq \mu_{(k)}) + \mathbb{I}(\mu_{(k)} < \mu_l). \end{aligned} \tag{A.3}$$

The μ_l for $l \in \mathbf{L}_k$ are independent given $\mu_{(1:k)}$, therefore we can obtain the distribution for $\mu_{(k+1)} = \max_{l \in \mathbf{L}_k} \mu_l$ by taking the product of the cdf's of μ_l's:

$$\begin{aligned} F(\mu_{(k+1)} \,|\, \mu_{(1:k)}) &= \big[\mu_{(k)}^{-\frac{\alpha}{K}} \mu_{(k+1)}^{\frac{\alpha}{K}} \mathbb{I}(0 \le \mu_{(k+1)} \le \mu_{(k)}) + \mathbb{I}(\mu_{(k)} < \mu_{(k+1)})\big]^{K-k} \\ &= \mu_{(k)}^{-\alpha\frac{K-k}{K}} \mu_{(k+1)}^{\alpha\frac{K-k}{K}} \mathbb{I}(0 \le \mu_{(k+1)} \le \mu_{(k)}) + \mathbb{I}(\mu_{(k)} < \mu_{(k+1)}). \end{aligned} \tag{A.4}$$

Differentiating the above equation, the density of $\mu_{(k+1)}$ is obtained to be,

$$\begin{aligned} p(\mu_{(k+1)} \,|\, \mu_{(1:k)}) &= p(\mu_{(k+1)} \,|\, \mu_{(k)}) \\ &= \alpha \frac{K-k}{K} \mu_{(k)}^{-\alpha\frac{K-k}{K}} \mu_{(k+1)}^{\alpha\frac{K-k}{K}-1} \mathbb{I}(0 \le \mu_{(k+1)} \le \mu_{(k)}). \end{aligned} \tag{A.5}$$

A.2 Probability of a part of Z being inactive

Given $\mu_{(k)}$, we can calculate the probability of all entries to the right of column k being zero. We denote the set of indices after the kth largest feature presence probability with $\mathbf{L}_k$. The density of the (unordered) feature presence probabilities with index $l \in \mathbf{L}_k$ is given in eq. (A.2). The entries z_{il} are Bernoulli distributed with probability μ_l. Marginalizing over μ_l, we have;

$$\begin{aligned} p(Z_{(:,.>k)} = 0 \,|\, \mu_{(k)}) &= \int p(Z_{(:,.>k)} = 0 \,|\, \mu_{(k)}, \mu_{\mathbf{L}}) p(\mu_{\mathbf{L}}) \mathrm{d}\mu_{\mathbf{L}} \\ &= \Big[\int_0^{\mu_{(k)}} \frac{\alpha}{K} \mu_{(k)}^{-\frac{\alpha}{K}} \mu^{\frac{\alpha}{K}-1} (1-\mu)^N \mathrm{d}\mu\Big]^{K-k} \end{aligned} \tag{A.6}$$

Applying change of variables $\nu = \mu/\mu(k)$ to the above integral,

$$\begin{aligned} &\int_0^{\mu_{(k)}} \frac{\alpha}{K} \mu_{(k)}^{-\frac{\alpha}{K}} \mu^{\frac{\alpha}{K}-1} (1-\mu)^N \mathrm{d}\mu \\ =&\int_0^1 \frac{\alpha}{K} \mu_{(k)}^{-\frac{\alpha}{K}} (\nu\mu_{(k)})^{\frac{\alpha}{K}-1} (1-\nu\mu_{(k)})^N \mu_{(k)} \mathrm{d}\nu \\ =&\int_0^1 \frac{\alpha}{K} \nu^{\frac{\alpha}{K}-1} (1-\nu+\nu-\nu\mu_{(k)})^N \mathrm{d}\nu \\ =&\frac{\alpha}{K} \int_0^1 \nu^{\frac{\alpha}{K}-1} \big((1-\nu)+\nu(1-\mu_{(k)})\big)^N \mathrm{d}\nu. \end{aligned} \tag{A.7}$$

Using the binomial series,

$$\begin{aligned} =&\frac{\alpha}{K} \int_0^1 \nu^{\frac{\alpha}{K}-1} \sum_{i=0}^N \binom{N}{i} (1-\nu)^{N-i} (\nu(1-\mu_{(k)}))^i \mathrm{d}\nu \\ =&\frac{\alpha}{K} \sum_{i=0}^N \binom{N}{i} (1-\mu_{(k)})^i \int_0^1 \nu^{i+\frac{\alpha}{K}-1} (1-\nu)^{N-i} \mathrm{d}\nu. \end{aligned} \tag{A.8}$$

Using the Dirichlet integral,

$$\begin{aligned}
&=\frac{\alpha}{K}\sum_{i=0}^{N}\binom{N}{i}(1-\mu_{(k)})^{i}\,\frac{\Gamma(i+\frac{\alpha}{K})\Gamma(N-i+1)}{\Gamma(N+\frac{\alpha}{K}+1)}\\
&=\frac{\alpha}{K}\sum_{i=0}^{N}\frac{N!}{(N-i)!\,i!}\,(1-\mu_{(k)})^{i}\,\frac{(N-i)!\prod_{j=0}^{i-1}\frac{\alpha}{K}+j}{\prod_{j=0}^{N}\frac{\alpha}{K}+j}\\
&=\frac{N!}{\prod_{j=1}^{N}\frac{\alpha}{K}+j}\,\Big\{1+\frac{\alpha}{K}\sum_{i=1}^{N}(1-\mu_{(k)})^{i}\,\frac{\prod_{j=1}^{i-1}\frac{\alpha}{K}+j}{i!}\Big\}.
\end{aligned} \tag{A.9}$$

Raising the above equation to the power $K-k$ and taking the limit as $K\rightarrow\infty$,

$$p(Z_{(:,.>k)}=0\,|\,\mu_{(k)})=\exp\big\{-\alpha H_N+\alpha\sum_{i=1}^{N}\frac{(1-\mu_{(k)})^{i}}{i}\big\}. \tag{A.10}$$

The first term is obtained by using the limit given in eq. (B.14), and the second term is obtained by the fact that the product inside the sum in the last line of eq. (A.9) reduces to $(i-1)!$ in the limit.

Using the series expansion for the natural logarithm given in eq. (B.16), the second term in the above equation can be approximated with a logarithm for large N,

$$\begin{aligned}
p(Z_{(:,.>k)}=0\,|\,\mu_{(k)})=&\exp\big\{-\alpha H_N-\alpha\ln(1-(1-\mu_{(k)}))\big\}\\
=&\mu_{(k)}^{-\alpha}\exp\big\{-\alpha H_N\big\}.
\end{aligned} \tag{A.11}$$

B Mathematical Appendix

B.1 Dirichlet Distribution

In the following, we describe the Dirichlet distribution and give some of its properties important for understanding the Dirichlet processes.

The *gamma* distribution with shape parameter $\alpha \geq 0$ and scale parameter $\beta > 0$ is given as

$$\mathcal{G}(\theta \,|\, \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} \theta^{\alpha-1} \exp\{-\theta/\beta\}. \tag{B.1}$$

Let θ_i, $i = 1, \dots, k$ be independent random variables with

$$\theta_i \sim \mathcal{G}(\alpha_i, 1). \tag{B.2}$$

The *Dirichlet* distribution $\mathcal{D}(\alpha_1, \dots, \alpha_k)$ is defined as the distribution of $(\pi_1, \dots, \pi_k)$, where

$$\pi_i = \theta_i / \sum_{j=1}^{k} \theta_j \quad \text{for} \quad i = 1, \dots, k. \tag{B.3}$$

The density function is given as,

$$\mathcal{D}(\pi_1, \dots, \pi_k \,|\, \alpha_1, \dots, \alpha_k) = \frac{\Gamma(\sum_{j=1}^{k} \alpha_j)}{\prod_{j=1}^{k} \Gamma(\alpha_j)} \prod_{j=1}^{k} \pi_j^{\alpha_j - 1}. \tag{B.4}$$

Dirichlet distribution is conjugate to the multinomial distribution

A random variable taking two different values with probability π_1 and $\pi_2 = 1 - \pi_1$ is *Bernoulli* distributed:

$$p(x) = \begin{cases} \pi_1, & x = x_1 \\ \pi_2, & x = x_2 \end{cases} \tag{B.5}$$

The count of the occurrence of one of the events (e.g. $x = x_1$) in a sequence of n Bernoulli trials is generally represented with the *Binomial* distribution:

$$p(y \,|\, \pi_1) = \mathrm{Bin}(y \,|\, \pi_1, n) = \binom{n}{y} \pi_1^{y} (1 - \pi_1)^{n-y}$$

The conjugate prior for the Binomial distribution is the *Beta* distribution:

$$p(\pi_1) = \mathrm{Beta}(\pi_1 \,|\, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi_1^{\alpha-1} (1 - \pi_1)^{\beta-1}.$$

Multinomial distribution is the distribution of a random variable that can take a countable number of different values. The conjugate prior for the multinomial distribution is the *Dirichlet* distribution, which is the multivariate generalization of the Beta distribution.

The marginal distribution of each π_i is beta

It follows from the definition of the Dirichlet distribution and the additive property of the gamma distribution that if $(\pi_1, \ldots, \pi_k) \sim \mathcal{D}(\alpha_1, \ldots, \alpha_k)$ and $r_1, \ldots, r_l$ are integers such that $0 < r_1, \ldots, r_l = k$, then

$$\mathcal{D}(\sum_{1}^{r_1} \pi_1, \sum_{r_1+1}^{r_2} \pi_i \ldots, \sum_{r_{l-1}+1}^{r_l} \pi_i \,|\, \alpha_1, \ldots, \alpha_k) \sim \mathcal{D}(\sum_{1}^{r_1} \alpha_1, \sum_{r_1+1}^{r_2} \alpha_i \ldots, \sum_{r_{l-1}+1}^{r_l} \alpha_i), \tag{B.6}$$

In particular, the marginal distribution of each π_i is $\mathrm{Beta}\big(\alpha_i, (\sum_1^k \alpha_j) - \alpha_i\big)$.

Scale of the Dirichlet distribution

The first and second moments of π_j are given by;

$$\mathrm{E}\{\pi_\mathrm{j}\} = \frac{\alpha_j}{\boldsymbol{\alpha}}, \qquad \mathrm{Var}(\pi_i, \pi_j) = \frac{\alpha_j(\boldsymbol{\alpha} - \alpha_j)}{\boldsymbol{\alpha}^2(\boldsymbol{\alpha} + 1)},$$

where $\boldsymbol{\alpha} = \sum_{j=1}^{K} \alpha_j$ is referred to as the scale of the distribution. The mean of the distribution does not depend on the scale of the parameters, however scale determines the spread.

Posterior distribution

The posterior distribution of the π_j given multinomially distributed data is:

$$p(\pi_1, \ldots, \pi_k \,|\, x_1, \ldots, x_n) = \mathcal{D}(\alpha_1 + n_1, \ldots, \alpha_k + n_k),$$

where n_j is the number of occurrence of the j^{th} event. The parameters of the Dirichlet distribution can be interpreted as the *pseudo* observations; for larger scale, the distribution will be more concentrated around the mean, thus the prior will have more effect on the posterior.

We can marginalize out the multinomial parameters π_j using the Dirichlet integral (B.11) and express the probability of the observation directly in terms of the Dirichlet parameters:

$$p(x_1, \ldots, x_n \,|\, \alpha_1, \ldots, \alpha_k) = \frac{\Gamma(\boldsymbol{\alpha})}{\Gamma(\sum_{l=1}^{k} \alpha_l + n)} \prod_{j=1}^{k} \frac{\Gamma(\alpha_j + n_j)}{\Gamma(\alpha_j)}$$

Thus, the conditional distribution for a new observation x_{n+1} given the previous observations is:

$$p(x_{n+1} = j \,|\, x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_k) = \frac{\alpha_j + n_j}{\sum_{i=1}^{k} \alpha_i + n}.$$

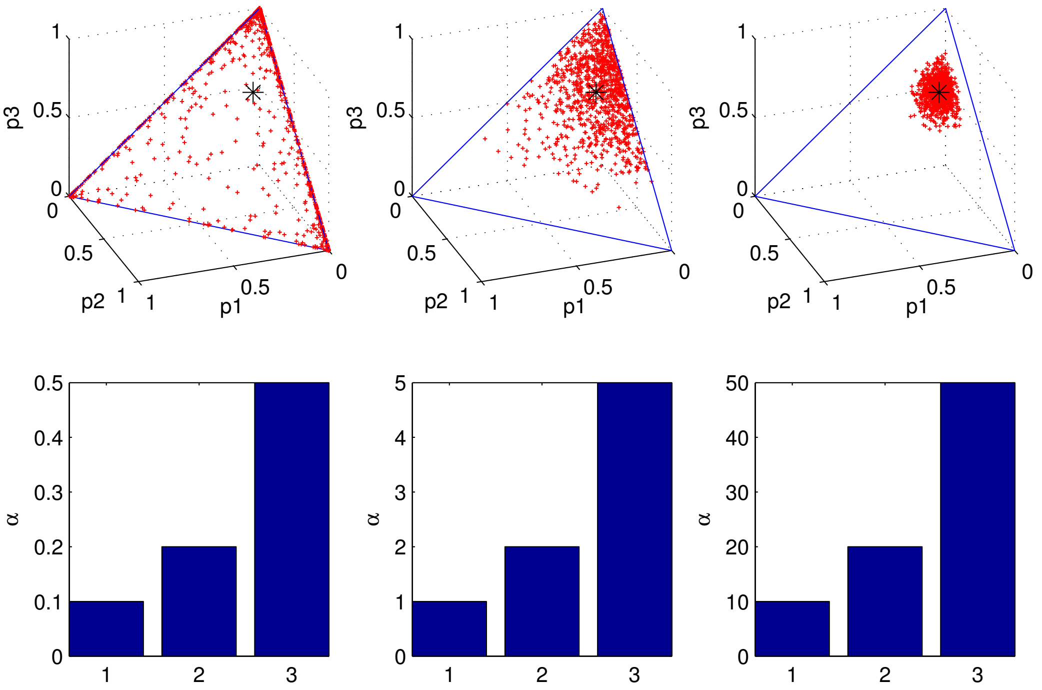

Figure B.1: Effect of the scale of the parameters for the Dirichlet distribution. Top row: Samples from Dirichlet distribution with three different parameter settings. Bottom row: Parameter values. The samples lie on the 2-D simplex denoted by the triangle. Note that the relative magnitudes for the distribution parameters are the same, however the scale changes, which effects the spread of the samples. The samples get more concentrated around the mean the higher the scale gets.

B.2 Parameterization of the Distributions Used

$$
\begin{array}{rl}
\mathrm{Beta}(\theta \,|\, \alpha, \beta) & : \ \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} \\
\mathrm{Binomial}(y \,|\, \pi_1, n) & : \ \dbinom{n}{y} \pi_1^y (1-\pi_1)^{n-y} \\
\mathrm{Dirichlet}(\pi_1, \ldots, \pi_k \,|\, \alpha_1, \ldots, \alpha_k) & : \ \dfrac{\Gamma(\sum_{j=1}^k \alpha_j)}{\prod_{j=1}^k \Gamma(\alpha_j)} \displaystyle\prod_{j=1}^k \pi_j^{\alpha_j - 1} \\
\mathrm{Gamma}(\theta \,|\, \alpha, \beta) & : \ \dfrac{1}{\Gamma(\alpha)\beta^\alpha} \theta^{\alpha-1} \exp\{-\theta/\beta\} \\
\mathrm{Multinomial}(\mathbf{y} \,|\, \boldsymbol{\pi}, n) & : \ \dbinom{n}{y_1\, y_2 \ldots y_k} \pi_1^{y_k} \ldots \pi_1^{y_k} \\
\mathrm{Normal}(\mathbf{x} \,|\, \boldsymbol{\mu}, \Sigma) & : \ \dfrac{1}{\sqrt{2\pi|\Sigma|}} \exp\big\{ -\dfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu}) \big\} \\
\mathrm{Poisson}(k \,|\, \lambda) & : \ \dfrac{\lambda^k \mathrm{e}^{-\lambda}}{k!} \\
\mathrm{Wishart}(W \,|\, S, \beta) & : \ \Big(2^{(\beta D/2)} \pi^{D(D-1)/4} \displaystyle\prod_{i=1}^D \Gamma(\dfrac{\beta+1-i}{2}) \Big)^{-1} \\
 & \quad \times |S|^{-\beta/2} \, |W|^{(\beta-D-1)/2} \exp\{-\dfrac{1}{2} \, \mathrm{tr}(S^{-1} \, W)\}
\end{array}
$$

B.3 Definitions

Lévy Processes

A distribution F is *infinitely divisible* if for every $n \geq 1$, there exists a characteristic function ψ_n such that

$$\psi_F(t) = \big(\psi_n(t)\big)^n, \tag{B.7}$$

that is, if F is the distribution of a sum of n i.i.d random variables.

Each infinitely divisible distribution corresponds to a process with stationary independent increments, called a *Lévy process.* Lévy-Khintchine formula states that every infinitely divisible distribution has a characteristic function $\psi(t) = \int \mathrm{e}^{itx} F(\mathrm{d}x)$ of the form,

$$\psi(t) = \exp\big\{ita \, - \, t^2\sigma^2/2 \, + \int [\mathrm{e}^{itu} - 1 - ith(u)] \, \nu(\mathrm{d}u)\big\}, \tag{B.8}$$

for $a \in \mathcal{R}$, $\sigma^2 \in \mathcal{R}_+$ and a Lévy measure $\nu(\mathrm{d}u)$.

A *subordinator* is a Lévy process with increasing sample paths.

Construction of A Process

The most fundamental characteristic of a random process is the set of its finite-dimensional distribution functions. Kolmogorov's theorem on existence of a process is the condition that needs to be satisfied for a family of distribution functions to define a process, see for example Shiryaev (1995).

Exchangeability

The sequence of random variables $\{\mathbf{x}_i\}_1^n$ is said to be exchangeable if any permutation $\{\mathbf{x}_{\pi_i},\ i = 1, \ldots, n\}$ have the same joint probability distribution. The variables of an infinite sequence are exchangeable if any finite subset of the sequence is exchangeable.

de Finetti's representation theorem:
If the sequence of random variables $\{\mathbf{x}_i\}_1^n$ is said to be infinitely exchangeable with probability measure P, there exists a probability measure Q over the space of all distribution functions on $\mathcal{R}$ such that the joint distribution function of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ has the following form:

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \int \big(\prod_{i=1}^{n} F(\mathbf{x}_i)\big)\, \mathrm{d}Q(F), \tag{B.9}$$

where $Q(F) = \lim_{n\to\infty} P(F_n)$, and F_n is the empirical distribution function defined by $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

B.4 Equalities and Limits

Recursive Property of the Gamma Function

The Gamma function has the following recursive property,

$$\Gamma(n) = (n-1)\Gamma(n-1), \tag{B.10}$$

For integer n, $\Gamma(n) = (n-1)!$.

The Dirichlet Integral

$$\int \theta_1^{\alpha_1-1} \ldots \theta_n^{\alpha_n-1} (1-\theta_1-\cdots-\theta_n)^{\alpha_{n+1}-1} \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_n = \frac{\prod_{i=1}^{n+1}\Gamma(\theta_i)}{\Gamma\big(\sum_{i=1}^{n+1}\theta_i\big)} \tag{B.11}$$

Poisson distribution as a limit of the beta distribution

The joint probability of sampling r of K new features for object i is

$$P(r \mid \alpha, K) = \left(\begin{array}{c} K \\ r \end{array}\right) \left(\frac{\alpha/K}{i+\alpha/K}\right)^r \left(1 - \frac{\alpha/K}{i+\alpha/K}\right)^{K-r}. \tag{B.12}$$

Taking the limit,

$$\begin{aligned}
&\lim_{K\to\infty} \begin{pmatrix} K \\ r \end{pmatrix} \left(\frac{\alpha/K}{i+\alpha/K}\right)^{r} \left(1-\frac{\alpha/K}{i+\alpha/K}\right)^{K-r} \\
&= \lim_{K\to\infty} \frac{K!}{(K-r)!r!} \frac{(\alpha/i)^r}{(K+\alpha/i)^r} \left(1-\frac{\alpha/i}{K+\alpha/i}\right)^{K} \left(1-\frac{\alpha/i}{K+\alpha/i}\right)^{-r} \\
&= \lim_{K\to\infty} \frac{K(K-1)\dots(K-r)}{(K+\alpha/i)^r} \frac{(\alpha/i)^r}{r!} \left(1-\frac{\alpha/i}{K+\alpha/i}\right)^{K} \left(1-\frac{\alpha/i}{K+\alpha/i}\right)^{-r} \\
&= \frac{(\alpha/i)^r \exp\{\alpha/i\}}{r!}.
\end{aligned} \tag{B.13}$$

The last equality is obtained from the fact that the limit of the first and the last term is 1 and

$$\lim_{K\to\infty} \left(\frac{1}{1+x/K}\right)^{K} = \exp\{-x\}. \tag{B.14}$$

Binomial Series

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} \tag{B.15}$$

Series expansion for the natural logarithm

$$\ln(1+z) = z - \frac{1}{2}z^2 + \frac{1}{3}z^3 - \dots, \quad |z| \leq 1, \text{and} z \neq 1 \tag{B.16}$$

Bibliography

D. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII–1983*, pages 1–198. Springer, Berlin, 1985.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

R. Batley and A. Daly. On the equivalence between elimination-by-aspects and generalised extreme value models of choice behaviour. *Journal of Mathematical Psychology*, 50:456–467, 2006.

R. R. Batsell, J. C. Polking, R. D. Cramer, and C. M. Miller. Useful mathematical relationships embedded in Tversky's elimination by aspects model. *Journal of Mathematical Psychology*, 47:538–544, 2003.

M. J. Beal and P. Krishnamurthy. Clustering gene expression time course data with countably infinite hidden markov models. In *Proceedings of UAI*, volume 22, 2006.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. In Z. G. T. Dietterich, S. Becker, editor, *Advances in Neural Information Processing Systems*, volume 15, pages 577–584. MIT Press, 2003.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory.* Chichester: Wiley, 1994.

D. Blackwell and J. B. MacQueen. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1:353–355, 1973.

D. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, Cambridge, MA, 2004.

D. M. Blei and M. I. Jordan. Variational methods for the Dirichlet process. *Journal of Bayesian Analysis*, 1(1):121–144, 2005.

G. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis.* John Wiley & Sons, 2003.

R. Bradley and M. Terry. The rank analysis of incomplete block designs. I. the method of paired comparisons. *Biometrika*, 39:324–345, 1952.

C. A. Bush and S. N. MacEachern. A semiparametric Bayesian model for randomised block designs. *Biometrika*, 83(2):275–285, 1996.

O. Cappé, C. P. Robert, and T. Rydén. Reversible jump, birth-and-death, and more general continuous time mcmc samplers. *Journal of the Royal Statistical Society, B*, 65:679–700, 2003.

W. Chu, Z. Ghahramani, R. Krause, and D. Wild. Identifying protein complexes in high-throughput protein interaction screens using an infinite latent feature model. In A. et al, editor, *BIOCOMPUTING 2006: Proceedings of the Pacific Symposium*, 2006.

H. Daumé III and D. Marcu. A Bayesian model for supervised clustering with the Dirichlet process prior. *Journal of Machine Learning Research*, 6:1551–1577, 2005.

D. Dey, P. Müller, and D. Sinha, editors. *Practical Nonparametric and Semiparametric Bayesian Statistics*. New York: Springer Verlag, 1998.

P. Diaconis and D. Freedman. On consistency of Bayes estimates. *The Annals of Statistics*, 14:1–67, 1986.

K. Doksum. Tailfree and neutral random probabilities and their posterior distributions. *The Annals of Probability*, 2(2):183–201, 1974.

A. Dubey, S. Hwang, C. Rangel, C. E. Rasmussen, Z. Ghahramani, and D. L. Wild. Clustering protein sequence and structure space with infinite Gaussian mixture models. In *Pacific Symposium on Biocomputing*, pages 399–410, Singapore, 2004. World Scientific Publishing.

M. D. Escobar. *Estimating the Means of Several Normal Populations by Estimating the Distribution of the Means*. PhD thesis, Yale University, 1988.

M. D. Escobar. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

M. D. Escobar and M. West. Computing Bayesian nonparametric hierarchical models. In D. Dey, P. Müller, and D. Sinha, editors, *Practical Nonparametric and Semiparametric Bayesian Statistics*, pages 1–22. New York: Springer Verlag, 1998.

E. V. Evarts. ique for recording activity of subcortical neurons in moving animals. *Electroencephalography and Clinical Neurophysiology*, 24(1):83–86, 1968.

P. Fearnhead. Particle filters for mixture models with an unknown number of components. *Statistics and Computing*, 14(1):11–21, 2004.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230, 1973.

T. S. Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4):615–629, 1974.

T. S. Ferguson and M. J. Klass. A representation of independent increment processes without gaussian components. *The Annals of Mathematical Statistics*, 43(5):1634–1643, 1972.

T. S. Ferguson, E. G. Phadia, and R. C. Tiwari. Bayesian nonparametric inference. In M. Gosh and D. Basu, editors, *Current Issues in Statistical Inference: Essays in Honor of D. Basu*, pages 127–150. Institute of Mathematical Statistics, 1992.

R. A. Fisher. The use of multiple measurements in axonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

E. Fokoué and D. M. Titterington. Mixtures of factor analysers: Bayesian estimation and inference by stochastic simulation. *Machine Learning*, 50:73–94, 2003.

M. Forina, C. Armanino, M. Castino, and M. Ubigli. Multivariate data analysis as a discriminating method of the origin of wines. *Vitis*, 25:189–201, 1986.

D. Freedman. On the asymptotic behaviour of Bayes estimates in the discrete case. *Annals of Mathematical Statistics*, 34(4):1386–1403, 1963.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, 2003.

Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixture of factor analysers. In A. Solla, T. K. Leen, and K. Mueller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 449–455. MIT Press, 2000.

Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996.

S. Ghosal, J. K. Ghosh, and R. V. Ramamoorth. Posterior consistency of Dirichlet mixtures in density estimation. *Annals of Statistics*, 27(1):143–158, 1999.

S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41:337–348, 1992.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice (Interdisciplinary Statistics)*. Chapman & Hall/CRC, 1995.

S. Goldwater, T. L. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power-law generators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.

D. Görür, C. E. Rasmussen, A. Tolias, F. Sinz, and N. Logothetis. Modelling spikes with mixtures of factor analysers. In C. E. Rasmussen, H. H. Buelthoff, M. A. Giese, and B. Schoelkopf, editors, *Pattern Recognition, Proc. 26th DAGM Symposium*, pages 391–398. Springer, 2004.

D. Görür, F. Jäkel, and C. E. Rasmussen. A choice model with infinitely many latent features. In *Proceedings of ICML*, volume 23, 2006.

J. K. Gosh and R. V. Ramamoorthi. *Bayesian Nonparametrics*. New York: Springer Verlag, 2003.

P. Green. Reversible jump Markov chain Monte Carlo Computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

P. Green and S. Richardson. Modelling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics*, 28:355–375, 2001.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. Technical Report 2005-01, Gatsby Computational Neuroscience Unit, University College London, 2005.

N. L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, 18(3):1259–1294, 1990.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, March 2001.

H. Ishwaran and L. F. James. Approximate Dirichlet process computing in finite normal mixtures: Smoothing and prior information. *Journal of Computational and Graphical Statistics*, 11(3):1–26, 2002.

H. Ishwaran and L. F. James. Computational methods for multiplicative intensity models using weighted gamma processes: Proportional hazards, marked point processes, and panel count data. *Journal of the American Statistical Association*, 99(465):175–190, 2004.

H. Ishwaran and G. Takahara. Independent and identically distributed Monte Carlo algorithms for semiparametric linear mixed models. *Journal of the American Statistical Association*, 97:1154–1166, 2002.

H. Ishwaran and M. Zarepour. Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

S. Jain and R. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 2000.

S. Jain and R. M. Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. Technical Report 0507, Department of Statistics, University of Toronto, 2005.

L. F. James and J. W. Lau. Flexible choice modelling based on Bayesian nonparametric mixed multinomial logit choice models. *Submitted*, 2004.

J. F. C. Kingman. Random discrete distributions. *Journal of the Royal Statistical Society, B*, 37:1–22, 1975.

A. Kottas and A. E. Gelfand. Bayesian semiparametric median regression modeling,. *Journal of the American Statistical Association*, 96:1458–1468, 2001.

K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 20, 2007a.

K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational Dirichlet process mixtures. In B. Sch´olkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007b.

M. Kuss, F. Jäkel, and F. A. Wichmann. Bayesian inference for psychometric functions. *Journal of Vision*, 5:478–492, 2005.

M. S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4):R53–R78, 1998.

J. S. Liu. Nonparametric hierarchical bayes via sequential imputations. *The Annals of Statistics*, 24:911–930, 1996.

H. F. Lopes and M. West. Bayesian model assessment in factor analysis. *Statistica Sinica*, 14:41–67, 2004.

R. Luce. *Individual Choice Behavior.* Wiley, New York, 1959.

S. MacEachern and P. Müller. Efficient MCMC schemes for robust model extensions using encompassing Dirichlet process mixture models. In F. Ruggeri and D. R. Insua, editors, *Robust Bayesian Analysis*, pages 295–315. Springer-Verlag, 2000.

S. N. MacEachern. Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics: Simulation and Computation*, 23:727–741, 1994.

S. N. MacEachern and P. Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7:223–238, 1998.

S. N. MacEachern, M. Clyde, and J. S. Liu. Sequential importance sampling for nonparametric Bayes models: The next generation. *The Canadian Journal of Statistics*, 27(2):251–267, 1999.

D. J. C. MacKay. Bayesian non-linear modeling for the energy prediction competition. *ASHRAE Transactions*, 100:1053–1062, 1994.

J. D. McAuliffe, D. M. Blei, and M. I. Jordan. Nonparametric empirical Bayes for the Dirichlet process mixture model. *Statisitcs and Computing*, 16:5–14, 2006.

D. McFadden. Economic choice. In T. Persson, editor, *Nobel Lectures, Economics 1996-2000*, pages 330–364. World Scientific Publishing, 2000.

B. L. McNaughton, J. O'Keffe, and C. A. Barnes. The stereotrode - a new technique for simultaneous isolation of several single units in the central nervous-system from multiple unit records. *Journal of Neuroscience Methods*, 8(4):391–397, 1983.

E. Meeds and S. Osindero. An alternative infinite mixture of gaussian process experts. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 883–890. MIT Press, Cambridge, MA, 2006.

E. Meeds, Z. Ghahramani, R. M. Neal, and S. T. Roweis. Modeling dyadic data with binary latent factors. In B. Sch´olkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

T. Minka and Z. Ghahramani. Expectation propagation for infinite mixtures. In *NIPS Workshop on Nonparametric Bayesian Methods and Infinite Models*, 2003.

P. Muliere and L. Tardella. Approximating distributions of random functionals of Ferguson-Dirichlet priors. *The Canadian Journal of Statistics*, 26(2):283–297, 1998.

P. Müller and F. A. Quintana. Nonparametric Bayesian data analysis. *Statistical Science*, 19(1):95–110, 2004.

D. Navarro and T. L. Griffiths. A nonparametric Bayesian method for inferring features from similarity judgment. In B. Sch´olkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

D. J. Navarro, T. L. Griffiths, M. Steyvers, and M. D. Lee. Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50:101–122, 2006.

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

R. M. Neal. Defining priors for distributions using Dirichlet diffusion trees. Technical report, University of Toronto, 2001. Apperaed in Valencia.

R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

R. M. Neal. Bayesian mixture modeling. In C. R. Smith, G. J. Erickson, and P. O. Neudorfer, editors, *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, pages 197–211. Dordrecht: Kluwer Academic Publishers, 1992.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Statistics, University of Toronto, 1993.

R. M. Neal. *Bayesian Learning for Neural Networks.* Number 118 in Lecture Notes in Statistics. Springer-Verlag, 1996.

M. A. Newton and Y. Zhang. A recursive algorithm for nonparametric analysis with missing data. *Biometrika*, 86:15–26, 1999.

D. P. Nguyen, L. M. Frank, and E. N. Brown. An application of reversible-jump Markov chain Monte Carlo to spike classification of multi-unit extracellular recordings. *Network: Computation in Neural Systems*, 14:61–82, 2003.

A. O'Hagan. *Advanced Theory of Statistics: Bayesian Inference v. 2B (Kendall's Advanced Statistics Library).* Hodder Arnold, 1994.

O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. submitted, 2005.

S. Petrone and A. E. Raftery. A note on the Dirichlet porcess prior in Bayesian nonparametric inference with partial exchangeability. *Statistics & Probability Letters*, 36: 69–83, 1997.

J. Pitman. *Combinatorial Stochastic Processes Ecole d'Eté de Probabilités de Saint-Flour XXXII – 2002*, volume 1875 of *Lecture Notes in Mathematics.* Springer, 2006.

J. Pitman. Some developments of the Blackwell-MacQueen urn scheme. In T. S. Ferguson, L. S. Shapley, and J. B. MacQueen, editors, *Statistics, Probability and Game Theory; Papers in honor of David Blackwell*, volume 30 of *Lecture Notes-Monograph Series*, pages 245–267. Institute of Mathematical Statistics, Hayward, CA, 1996.

J. Pitman and M. Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.

I. Porteus, A. Ihler, P. Smyth, and M. Welling. Gibbs sampling for (coupled) infinite mixture models in the stick-breaking representation. In *Proceedings of UAI*, volume 22, 2006.

F. A. Quintana. Nonparametric bayesian analysis for assessing homogeneity in k × i contingency tables with fixed right margin totals. *Journal of the American Statistical Association*, 93(443):1140–1149, 1998.

C. E. Rasmussen. The infinite Gaussian mixture model. In S. A. Solla, T. K. Leen, and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 554–560. MIT Press, 2000.

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, Cambridge, MA, 2006.

M. L. Recce and J. O'Keefe. The tetrode: An improved technique for multi-unit extracellular recording. *Society for Neuroscience Abstracts*, 15(2):1250, 1989.

F. Restle. *Psychology of Judgment and Choice: A Theoretical Essay.* John Wiley & Sons, 1961.

S. Richardson and P. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, B*, 59:731–792, 1997.

S. Ruan, S. N. MacEachern, T. Otter, and A. M. Dean. The dependent poisson race model and modeling dependence in conjoint choice experiments. Technical Report 767, The Ohio State University, Department of Statistics, 2005.

D. Rumelhart and J. Greeno. Similarity between stimuli: An experimental test of the Luce and Restle choice models. *Journal of Mathematical Psychology*, 8:370–381, 1971.

D. W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization.* Wiley, 1992.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

J. Sethuraman and R. C. Tiwari. Convergence of Dirichlet measures and the interpretation of their parameter. In S. S. Gupta and J. O. Berger, editors, *Statistical Decision Theory and Related Topics, III*, volume 2, pages 305–315. Academic Press, London, 1982.

X. Shen and L. Wasserman. Rates of convergence of posterior distributions. *Annals of Statistics*, 29, 2001.

A. N. Shiryaev. *Probability (2nd ed.).* Springer-Verlag New York, Inc., 1995.

S. Sibisi and J. Skilling. Prior distributions on measure space. *Journal of the Royal Statistical Society, B*, 59(1):217–235, 1997.

K. Sohn and E. Xing. A hidden markov Dirichlet process model for genetic recombination in open ancestral space. In B. Sch´olkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

M. Stephens. Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods. *The Annals of Statistics*, 28: 40–74, 2000.

E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1297–1304. MIT Press, Cambridge, MA, 2006.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proceedings of AISTATS*, 2007.

R. Thibaux and M. I. Jordan. Hierarchical beta process and the Indian buffet process. In *Proceedings of AISTATS*, 2007.

K. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2003.

A. Tversky. Elimination by aspects: A theory of choice. *Psychological Review*, 79: 281–299, 1972.

A. Tversky and S. Sattath. Preference trees. *Psychological Review*, 86(6):542–573, 1979.

A. Utsugi and T. Kumagai. Bayesian analysis of mixtures of factor analyzers. *Neural Computation*, 13:993–1002, 2001.

S. Walker. Sampling the Dirichlet mixture model with slices. Technical Report 16, International Centre for Economic Research, 2006.

S. Walker and P. Damien. Sampling methods for Bayesian nonparametric inference involving stochastic processes. In D. Dey, P. Müller, and D. Sinha, editors, *Practical Nonparametric and Semiparametric Bayesian Statistics*, chapter 13, pages 243–254. Springer-Verlag, 1998.

S. Walker, P. Damien, P. W. Laud, and A. F. M. Smith. Bayesian nonparametric inference for random distributions and related functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61:485–527, 1999. (with discussion).

S. G. Walker, A. Lijoi, and I. Prünster. On rates of convergence for posterior distributions in infinite-dimensional models. *Annals of Statistics*, 2006.

M. West, P. Müller, and M. D. Escobar. Hierarchical priors and mixture models with applications in regression and density estimation. In P. R. Freeman and A. F. M. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.

F. Wickelmaier and C. Schmid. A Matlab function to estimate choice model parameters from paired comparison data. *Behavior Research Methods, Instruments, & Computers*, 36(1):29–40, 2004.

R. L. Wolpert and K. Ickstadt. Simulation of Lévy random fields. In D. Dey, P. Müller, and D. Sinha, editors, *Practical Nonparametric and Semiparametric Bayesian Statistics*, chapter 12, pages 227–242. Springer, 1998.

F. Wood, S. Goldwater, and M. J. Black. A non-parametric Bayesian approach to spike sorting. In *Proceedings of the 28th IEEE Conference on Engineering in Medicine and Biologicial Systems*, pages 1165–1169, 2006a.

F. Wood, T. L. Griffiths, and Z. Ghahramani. A non-parametric Bayesian method for inferring hidden causes. In *Proceedings of UAI*, volume 22, 2006b.

E. Xing, R. Sharan, and M. Jordan. Bayesian haplotype inference via the Dirichlet process. In *Proceedings of ICML*, volume 21, 2004.

E. P. Xing, K.-A. Sohn, M. I. Jordan, and Y. W. Teh. Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture. In *Proceedings of ICML*, volume 23, 2006.

Y. Xue, X. Liao, C. Lawrence, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 1617–1624. MIT Press, Cambridge, MA, 2005.