A deterministic inference framework for discrete nonparametric latent variable models

Learning complex probabilistic models with simple algorithms

Yordan P. Raykov

Doctor of Philosophy

ASTON UNIVERSITY

January 2017

Aston University

A deterministic inference framework for discrete nonparametric latent variable models

Learning complex probabilistic models with simple algorithms

Yordan P. Raykov

Doctor of Philosophy, 2017

Abstract

Latent variable models provide a powerful framework for describing complex data by capturing its structure with a combination of more compact unobserved variables. The Bayesian approach to statistical latent models additionally provides a consistent and principled framework for dealing with uncertainty inherent in the data described with our model. However, in most Bayesian latent variable models we face the limitation that the number of unobserved variables has to be specified a priori. With the increasingly larger and more complex data problems such parametric models fail to make most out of the data available. Any increase in data passed into the model only affects the accuracy of the inferred posteriors and models fail to adapt to adequately capture new arising structure. Flexible Bayesian nonparametric models can mitigate such challenges and allow the learn arbitrarily complex representations given enough data is provided. However, their applications are restricted to applications in which computational resources are plentiful because of the exhaustive sampling methods they require for inference.

At the same time we see that in practice despite the large variety of flexible models available, simple algorithms such as K-means or Viterbi algorithm remain the preferred tool for most real world applications. This has motivated us in this thesis to borrow the flexibility provided by Bayesian nonparametric models, but to derive easy to use, scalable techniques which can be applied to large data problems and can be ran on resource constraint embedded hardware.

We propose nonparametric model-based clustering algorithms nearly as simple as K-means which overcome most of its challenges and can infer the number of clusters from the data. Their potential is demonstrated for many different scenarios and applications such as phenotyping Parkinson and Parkisonism related conditions in an unsupervised way. With few simple steps we derive a related approach for nonparametric analysis on longitudinal data which converges few orders of magnitude faster than current available sampling methods. The framework is extended to efficient inference in nonparametric sequential models where example applications can be behaviour extraction and DNA sequencing. We demonstrate that our methods could be easily extended to allow for flexible online learning in a realistic setup using severely limited computational resources. We develop a system capable of inferring online nonparametric hidden Markov models from streaming data using only embedded hardware. This allowed us to develop occupancy estimation technology using only a simple motion sensor.

Keywords: Bayesian nonparametrics, clustering, segmentation, mixture models, hidden Markov models.

Acknowledgments

First, I would like to express my sincere gratitude to my supervisor Max Little. During the last three years, he has provided me with unparalleled training environment. I have truly enjoyed our thought provoking group meetings and I have learned more from them than from any book I have read or any class I have taken. Most of all I thank him for his passion and enthusiasm for research which has set an example for me to follow and has sweetened the countless efforts put towards completing this work.

I would also like to thank Alexis Boukouvalas for his mentorship, encouragement, patience and support. He has had a great input on many of the ideas included in this thesis and I can safely say that without him the PhD journey would not have been nearly as much fun as it was.

I am extremely grateful to Emre Ozer for the wonderful time I had being a part of his team in ARM Research Cambridge. He has given a great twist to my research focus and has helped me discover great deal of problems I wish I could solve in future.

I also feel lucky for being a part of the Nonlinearity and Complexity Research Group in Aston whose staff and students create a wonderful and stimulating research environment. Special thanks to David Saad, David Lowe and Ian Nabney for their valuable feedback on the work presented in this thesis. I also would like to thank John E Smith for honouring me with an award for my work towards this thesis.

My research has really benefited from the knowledge I gained during all the workshops, conferences and summers schools I visited. There I was amazed with how generous leaders in the field are with their time and knowledge. Short discussions with Erik Sudderth, Jim Griffin, Michael Jordan and Francois Caron have potentially saved years of my time trying to understand some of the issues related to Bayesian nonparametrics.

I owe a great debt to Peter for helping me fight the English grammar which is an old enemy of mine. I also really appreciate Reham's efforts on proofing my thesis. And I thank Bilyan for being the first reader of my first paper.

A big thanks to all of my friends at home and across the world, for being always ready to distract me. Better not to mention names as I am bound to miss someone.

I just have to thank Victoria for all of her support and patience in the last years. And last but certainly not least I owe the world to my mother for always being there for me. I also cannot overlook the impact she had on the contributions of this thesis because nothing makes you understand a subject, like having to explain it to your mother.

Dedicated to my mother and my late dad.

Contents

1 Introduction **12**

1.1 Motivation . . . 12
1.2 Contributions . . . 16
1.3 Thesis organization . . . 19

2 Discrete latent variable models and inference **21**

2.1 The K-means algorithm . . . 21
2.2 Mixture models . . . 25
2.3 The Bayesian framework . . . 26
2.3.1 Bayesian mixture models . . . 26
2.3.2 Gibbs sampling . . . 29
2.3.3 Variational Bayes inference . . . 30
2.3.4 Iterated conditional modes . . . 32
2.4 Marginalization . . . 33
2.5 The Dirichlet process . . . 34
2.5.1 Definition . . . 35
2.5.2 Constructions . . . 36
2.5.3 Pitman-Yor generalization . . . 39
2.5.4 Dirichlet process mixture models . . . 40
2.6 Overview of the relations between inference algorithms . . . 42

3 Simple deterministic inference for mixture models **45**

3.1 Introduction . . . 45
3.2 Small variance asymptotics . . . 46
3.2.1 Probabilistic interpretation of K-means . . . 46
3.2.2 K-means with reinforcement . . . 47
3.2.3 Overview . . . 48
3.3 Rao-Blackwellization in mixture models . . . 49
3.4 Collapsed K-means and collapsed MAP-GMM . . . 52
3.5 Comparison on synthetic data . . . 53
3.6 Nonparametric clustering alternatives . . . 55
3.6.1 Gibbs sampling for DPMM . . . 55
3.7 Deterministic inference for Dirichlet process mixtures . . . 57
3.7.1 Variational inference for DPMMs . . . 57
3.7.2 Small variance asymptotics methods for DPMMs . . . 59

3.8 Iterative maximum-a-posteriori inference . . . 62
3.8.1 Collapsed MAP-DPMM algorithm . . . 62
3.8.2 The MAP-DPMM algorithm . . . 64
3.8.3 Out-of-sample prediction . . . 65
3.8.4 Analysis of iterative MAP for DPMM . . . 66
3.9 DPMM experiments . . . 67
3.9.1 UCI experiment . . . 67
3.9.2 Synthetic CRP parameter estimation . . . 71
3.10 Example applications of MAP-DPMM algorithms . . . 73
3.10.1 Sub-typing of parkinsonism and Parkinson's disease . . . 73
3.10.2 Application of MAP-DPMM to semiparametric mixed effect models . . . 76
3.11 Discussion . . . 78

4 Deterministic inference and analysis of HDP mixtures 79
4.1 Introduction . . . 79
4.2 Motivation . . . 79
4.3 Hierarchical Dirichlet process . . . 80
4.3.1 Stick-breaking construction for the HDP . . . 80
4.3.2 Chinese restaurant franchise . . . 81
4.3.3 Gibbs sampling for HDP mixture models . . . 82
4.4 Deterministic inference for HDPs . . . 87
4.4.1 SVA inference for HDP mixtures . . . 88
4.4.2 Iterative maximum a-posteriori inference . . . 90
4.5 Synthetic study . . . 92
4.6 Discussion . . . 93

5 Model-based nonparametric analysis of sequential data 94
5.1 Introduction . . . 94
5.2 Hidden Markov models . . . 95
5.3 Nonparametric Bayesian HMM . . . 98
5.3.1 Gibbs sampling methods for the HDP-HMM . . . 99
5.4 Deterministic methods . . . 104
5.4.1 SVA analysis for HDP-HMM . . . 105
5.4.2 Iterative MAP inference for iHMM . . . 107
5.5 Applications . . . 111
5.5.1 Genomic hybridization and DNA copy number variation . . . 111
5.5.2 Behaviour extraction from accelerometer data . . . 113
5.6 Discussion . . . 115

6 Occupancy estimation using nonparametric HMMs 116
6.1 Introduction . . . 116
6.1.1 Motivation . . . 116
6.1.2 Challenges of human occupancy counting with a single PIR sensor . . . 117
6.1.3 Related work . . . 117
6.2 Experimental Setup . . . 118

6.2.1 Collection devices . . . 118
6.2.2 Data collection . . . 119
6.2.3 Sensor data description . . . 120
6.3 Laplace modeling . . . 121
6.3.1 Regression component . . . 121
6.3.2 Time window duration . . . 122
6.4 Extracting behaviour from PIR data . . . 123
6.5 System overview . . . 125
6.6 System evaluation . . . 125
6.6.1 Fewer than 8 occupants . . . 126
6.6.2 At least 8 occupants . . . 127
6.7 Computational efficiency . . . 127
6.7.1 Choice of inference algorithm . . . 127
6.7.2 Resource evaluation . . . 129
6.8 Future work . . . 131
6.9 Discussion . . . 132

7 Conclusion **133**
7.1 Summary . . . 133
7.2 Future directions . . . 134
7.2.1 Unsupervised behaviour modeling . . . 135
7.2.2 Real time learning . . . 135
7.2.3 Parallel iterative MAP methods . . . 136

A Hyper parameters updates for exponential family conjugate pairs **147**

B Implementation practicalities **152**
B.1 Randomized restarts . . . 152
B.2 Obtaining cluster centroids . . . 153

C Out-of-sample predictions **154**

D Missing data **155**

E Estimating the model hyper parameters (θ_0, N_0) **156**

F Bregman divergences **158**

G DP-means λ parameter binary search **159**

H Gibbs sampling for DPMM (spherical Gaussian) **160**

I Fully collapsed CRF-based Gibbs sampler **162**

J PD-DOC experiment **164**

List of Figures

1.1 Clustering spherical Gaussian data with varying density . . . 13
1.2 Clustering spherical Gaussian data with varying spread . . . 14
1.3 Clustering well-separated elliptical data with K-means and MAP-DP . . . 15
1.4 Clustering spherical Gaussian data with K-means and MAP-DP . . . 17
1.5 Clustering trivial elliptical Gaussian data with K-means and MAP-DP . . . 18
1.6 Clustering challenging Gaussian data with K-means and MAP-DP . . . 19

2.1 Probabilistic graphical model of the Bayesian mixture model . . . 28
2.2 Illustration of the Chinese Restaurant Process . . . 39
2.3 Synthetically generated data from DPMM . . . 42
2.4 Different approaches to inference in parametric and nonparametric mixture models. . . . 43
2.5 Different approaches to inference in parametric and nonparametric hidden Markov models. . . . 44

3.1 Association chart of SVA, ICM and Gibbs sampling . . . 49
3.2 Probabilistic graphical model of the collapsed Bayesian mixture model . . . 51
3.3 Density of the Student-t distribution with varying degrees of freedom . . . 68
3.4 CRP mixture experiment . . . 72
3.5 Identified cluster for the ELSA longitudinal data. . . . 77

4.1 Illustrating the hierarchical Dirichlet process (HDP) . . . 82
4.2 Graphical model of the HDP mixture . . . 83
4.3 Four of the synthetic data sets used for HDP experiments . . . 93

5.1 Graphical model for the Bayesian HMM. . . . 95
5.2 Convergence of Gibbs and beam samplers for HDP-HMM . . . 100
5.3 Reconstructing Gaussian HMM data with asymp-iHMM . . . 107
5.4 Convergence of MCMC and 'nearly' iterative MAP methods . . . 109
5.5 Reconstructing Gaussian HMM data with MAP methods . . . 111
5.6 MAP-iHMM applied for identification of DNA copy number regimes . . . 112
5.7 Distribution of accelerometer output from the gait for different tests. . . . 114
5.8 Data quality control using MAP-iHMM . . . 114

6.1 Collecting PIR data with embedded hardware . . . 119
6.2 Example of training enviroment . . . 119
6.3 Comparison of the PIR sensor output for occupied and for empty enviroment. . . . 120
6.4 Frequency plots of PIR data . . . 120

6.5 Modelling occupancy with Laplace distribution 121
6.6 Effect of window duration on estimated Laplace paramters 123
6.7 Illustration of iHMM applied to PIR data 124
6.8 Architecture of a novel occupancy estimation system. 125
6.9 Box plots of extimated Laplace parameters 127
6.10 Box plots of Laplace parameters after segmentation 128
6.11 Box plots of Laplace parameters estimated from overpopulated enviroment 129

List of Tables

3.1 Clustering performance of MAP, K-means, E-M and Gibbs on synthetic data 54
3.2 Iterations to convergence of MAP, K-means, E-M and Gibbs on synthetic data 54
3.3 Clustering performance of DPMM inference techniques for the wine dataset from the UCI machine learning repository 69
3.4 Clustering performance of DPMM inference techniques measured for the iris dataset from the UCI machine learning repository 69
3.5 Clustering performance of DPMM inference techniques measured for the breast cancer dataset from the UCI machine learning repository 69
3.6 Clustering performance of DPMM inference techniques measured for the soybean dataset from the UCI machine learning repository 70
3.7 Clustering performance of DPMM inference techniques measured for the Pima dataset from the UCI machine learning repository 70
3.8 Clustering performance of collapsed MAP-DPMM and collapsed Gibbs-DPMM sampling inference applied to Gaussian DPMM with complete covariances: UCI datasets 71
3.9 Performance of collapsed Gibbs-DPMM, collapsed MAP-DPMM, DP-means and VB-DPMM inference methods used for clustering synthethic DPMM distributed data 72
3.10 Significant features of parkinsonism from the PostCEPT/PD-DOC clinical reference data across clusters (groups) obtained using collapsed MAP-DPMM 74
3.11 Significant likert features of parkinsonism from the PostCEPT/PD-DOC clinical reference data across clusters obtained using collapsed MAP-DPMM 74
3.12 Cross-validated, average held-out likelihood for two models. 77

4.1 Comparison of SVA and MAP methods applied to HDP mixture 92

5.1 Comparison between MAP and MCMC methods for HDP-HMM 111

6.1 Accuracy of the estimated occupancy for up to 7 present 126
6.2 Accuracy of the estimated occupancy for more than 8 present 126
6.3 Efficiency of various iHMM inference methods for the task of occupancy estimation 128
6.4 Specification of the MCUs. 130
6.5 Computation time and memory consumption of the system ran on difference MCUs 130
6.6 Power consumption and battery lifetime 130

J.1 Binomial features PD-DOC dataset 165
J.2 Binary features PD-DOC dataset 166
J.3 Categorical features PD-DOC dataset 166

J.4 Poisson Data 166

Chapter 1

Introduction

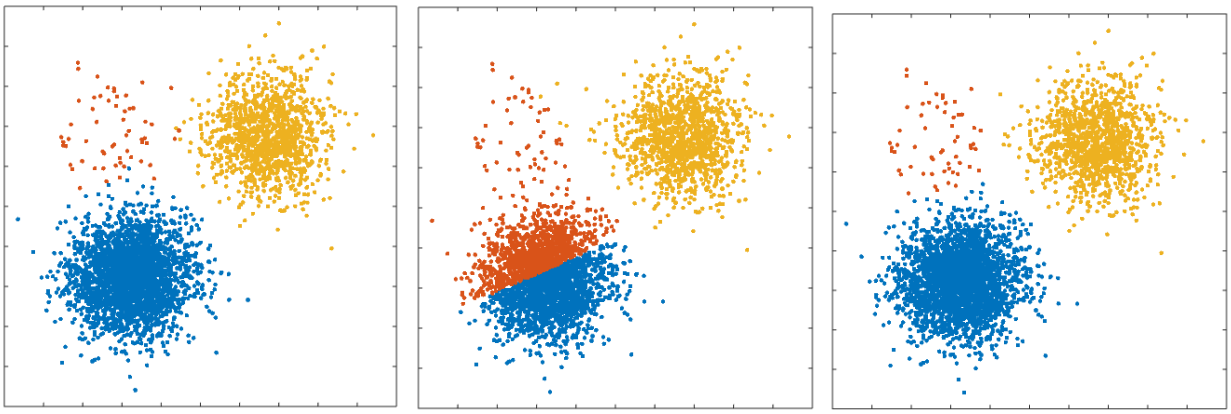
1.1 Motivation

The rapid increase in the capability of automatic data acquisition and storage is providing striking potential for innovation in science and technology. However, extracting meaningful information from complex, ever-growing data sources poses new challenges. This motivates the development of automated yet principled ways to discover structure in data. The key information of interest is often obscured behind redundancy and noise, therefore designing a plausible and statistical (or mathematical) model becomes challenging. Complex data models can be expressed in more tractable form if we instead model them using a combination of simpler components: for example consider the distribution of some data modeled with the joint distribution over an extended space consisting of both the observed variables and some *latent variables*. Latent variables describe some unobserved structure in the data the type of which we define through a set of encoded assumptions inherent to our model. For example, based on what type of latent variables we assume, latent variable models can be separated into two classes: discrete and continuous latent models.

Broadly speaking, continuous latent variable models are useful for problems where data lies close to a manifold of much lower dimensionality. By using continuous latent variables, we can express inherent unobserved structure (considering it does exist) in the data with significantly fewer latent variables and therefore these latent variable models play a key role in the statistical formulation of many *dimensionality reduction* techniques. Many widely-used pattern recognition techniques can be understood in that framework: probabilistic *principle component analysis* (PCA) (Tipping & Bishop, 1999; Roweis, 1998), the Kalman filter and others. In addition, as Tipping & Bishop (1999) have pointed out, many non-probabilistic methods can be well understood as a restricted case of a continuous variable model: *independent component analysis* and *factor analysis* for example Spearman (1904) which describe variability among observed, correlated variables.

By contrast, discrete latent variable models assume discreteness in the unobserved space. This discreteness naturally implies that random draws from this space have a finite probability of repetition which is one reason why discrete latent models are widely used to express inherent groupings and similarities that underlie the data. They have played a key role in the probabilistic formulation of *clustering* techniques. However, computationally inferring (learning) such models from the data is a lot more challenging than for continuous latent variable models and in its full generality clustering implies a combinatorial (NP-hard) problem. This often restricts the application of discrete latent models to applications in which computational resources and time for inference is plentiful.

In this thesis we will try to approach this problem by analyzing a rigorous framework for inference in



(a) Generated synthetic data (b) K-means (c) MAP-DP

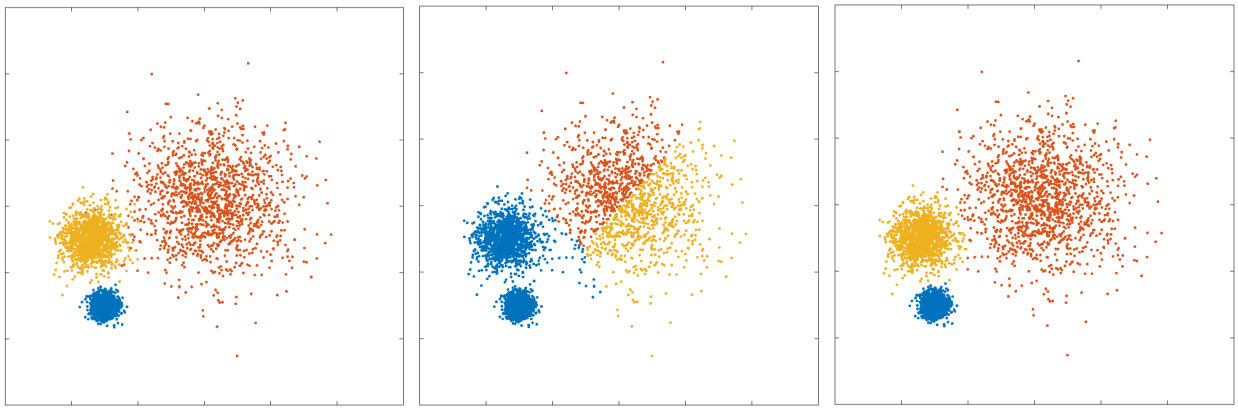
Figure 1.1: Clustering performed by K-means and MAP-DP for spherical, synthetic Gaussian data. Cluster radii are equal and clusters are well-separated, but the data is unequally distributed across clusters: 69% of the data is in the blue cluster, 29% in the yellow, 2% is orange. K-means fails to find a meaningful solution, because, unlike MAP-DP, it cannot adapt to different cluster densities, even when the clusters are spherical, have equal radii and are well-separated.

discrete latent variable models. To allow for more in-depth analysis of the techniques introduced in this thesis, we will focus only on a few models which are foundational models in the field of machine learning. This will allow us to make more explicit the benefits of the proposed framework and its specific applications without having to tackle the full complexity of the overwhelmingly rich class of discrete latent variable models. However, we note that a lot of the issues we discuss here can be extrapolated to more complex and elaborate models and therefore this should be viewed as a starting point for future work in this direction.

Optimizing the efficiency of our inference procedures is not enough on its own to handle the steady growth both in terms of size and complexity of data problems that we find ourselves facing in recent years. The striking increases in the amount of data available for statistical analysis suggests a need to also change the statistical models we use. Restrictive assumptions about the structure and the complexity of a model are less likely to hold and harder to define for such situations This introduces the need for more adaptable *Bayesian nonparametric* (BNP) models which can be used to relax such restrictions. We cannot fully appreciate the advantages that such models bring without first formally specifying the statistical and mathematical problem of *model selection*.

Model selection Determining the structure underlying some set of observations often translates to the problem of learning a mathematical model that can accurately predict those observations. In the case of probabilistic models, we often specify the particular model and we search through a set of parameter values in the model, so that the model best explains the observations according to some criteria. This criteria is designed to indicate the generalization of a model, or how well the model describes the population of the data rather than just the observed sample. Models that are too simple *underfit* the data and fail to capture all of the inherent structure in it; models which are too complex *overfit* suggesting structure for which there is insufficient evidence. For example, in the case of clustering an underfitted model would fail to discover all of the distinct clusters in the data where an overfitted model would suggest more clusters than there actually are.

A natural way to design models that are resilient to overfitting and directly enable adequate model selection is to adopt the Bayesian formalism. The Bayesian treatment to probabilistic models views all model



(a) Generated synthetic data (b) K-means (c) MAP-DP

Figure 1.2: Clustering performed by K-means and MAP-DP for spherical, synthetic Gaussian data, with unequal cluster radii and density. The clusters are well-separated. Data is equally distributed across clusters. Here, unlike MAP-DP, K-means fails to find the correct clustering. Instead, it splits the data into three equal-volume regions because it is insensitive to the differing cluster density. Different colours indicate the different clusters.

parameters as random variables. The distributions that specify them are called *prior* distributions and they provide additional control over the behaviour of the assumed model. Model selection and model comparison can be directly performed in Bayesian models by computing the *marginal likelihood* of the model. Specifically, in latent variable models the marginal likelihood is computed by taking the complete data likelihood function and integrating out the latent variables. The model which has the highest marginal likelihood is the model that best describes the data and provides the optimal fit. There are many other widely techniques for assessing the quality of fit of a model such as cross-validation, bootstraping, regularization or Bayes factors, to name a few. However, in the framework of probabilistic models many involve additional ad-hoc (heuristic) assumptions to the existing model which are not necessarily justified nor well understood. By contrast, the Bayesian paradigm addresses the specification of the model and the problem of overfitting at the same time (Bishop, 2013). This often makes Bayesian models more forgiving to differences between the specification of the model and the observed data.

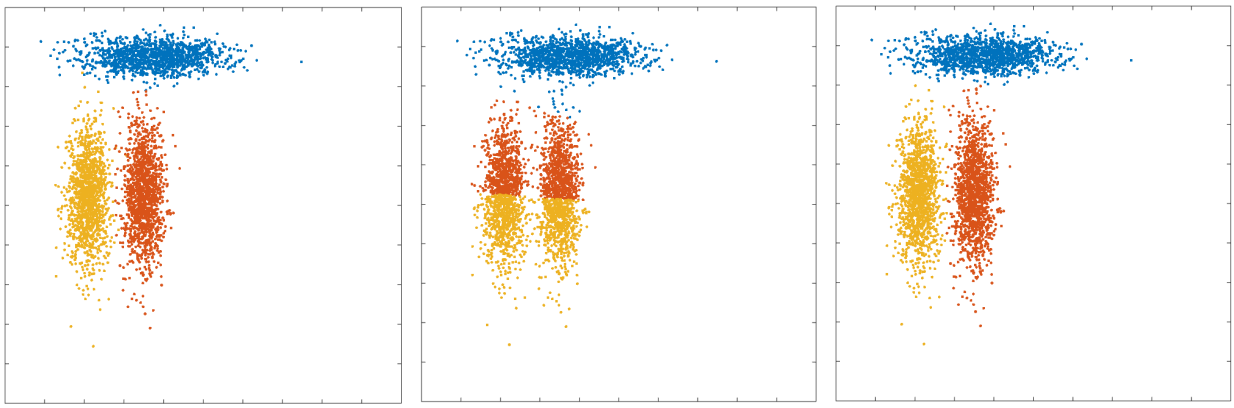
Now that we have defined the problem of model selection and the Bayesian approach to solving it, we can go back to specifying what we mean by BNP models in the context of latent variable models. A large class of probabilistic models (Bayesian and non-Bayesian) can be classed as *parametric*. Such models require the specification and choice of the number of model parameters: this is often an effective measure its complexity. In discrete latent variable models, this usually means that parametric models fix the domain of the latent variables; for example in clustering this implies fixing the number of clusters that can be found. BNP models relax this assumption allowing the model to adapt its complexity depending upon the data on which it is trained . The domain of the latent variables is defined as infinite meaning that the complexity of the unobserved structure can grow and adapt based on the evidence.

To give some specific examples, one of the most popular discrete latent variable models is the *Gaussian mixture model* (GMM) which is formally defined in Chapter 2. The GMM models the complete dataset with a mixture of Gaussian distributions which can express complex data distributions using a combination of simple Gaussians. The unobserved variables here indicate which particular Gaussian best describes each specific point from the data. In the parametric setting the number K of Gaussian distributions forming the likelihood of the GMM needs to be specified by design and it remains unchanged despite the size or structure

of the data . The BNP extension of the GMM (which we also define formally in Chapter 2) uses as many Gaussian distributions as considered sufficient according to the model likelihood. That is, the nonparametric nature of the model aims to keep it from underfitting and the Bayesian nature of the model makes it resilient to overfitting (Hjort *et al.*, 2010).

The problem of choosing K in a GMM has also been widely addressed outside of the Bayesian paradigmby the use of different *regularization* criteria: *Bayesian information criterion* (BIC)(Pelleg *et al.*, 2000); *minimum description length* (MDL)(Bischof *et al.*, 1999); *deviance information criterion* (DIC)(Gao *et al.*, 2011) to name a few. Typically a parametric model, like the GMM is fitted for different values of K and a regularization criterion is used to choose the value of K that provides the best overall fit of the model. This means repeating our inference algorithm multiple times to exhaustively search the space of K and also relying on additional assumptions about the model which are inherent to the reguralizer but not part of the model itself[1].

Unfortunately, in practice the flexibility and expressive power granted by BNP models (and often also of parametric models) carries a heavy computational price because they require computationally intensive inference methods such as Markov chain Monte Carlo sampling techniques (Hastings, 1970; Geman & Geman, 1984; Chib & Greenberg, 1995; Neal, 2000, 2003; Van Gael *et al.*, 2008). This is an emerging problem because we more and more often face the following situations: problems with large-scale datasets; "embedded" applications such as *Internet of Things* (IoT) devices where computation needs to be performed in real-time and countless applications (for example, digital signal processing; ubiquitous computing) where computation needs to be executed on resource-constrained hardware. We are witnessing the end of *Moore's law* (Schaller, 1997; Kish, 2002; Colwell, 2013) which has dominated the way we think about computing and computational algorithms over the last 50 years[2]. Thus, there is the increasingly pressing need for approaches to inference that are not only accurate but also use minimal computational effort (Bousquet & Bottou, 2008).



(a) Generated synthetic data (b) K-means (c) MAP-DP

Figure 1.3: Clustering solution obtained by K-means and MAP-DP for synthetic elliptical Gaussian data. All clusters share exactly the same volume and density, but one is rotated relative to the others. There is no appreciable overlap. K-means fails because the objective function which it attempts to minimize measures the true clustering solution as worse than the manifestly poor solution shown here.

[1]We demonstrate many of the disadvantages of regularization techniques applied to GMM and K-means clustering algorithm in (Raykov *et al.*, 2016c).

[2]Moore's law refers to an observation made by Intel co-founder Gordon Moore in 1965. Moore's law predicts that the number of transistors per square inch on integrated circuits would double every year into the foreseeable future. In 1975 Gordan Moore revisited his forecast to doubling every two years. This forecast has been true for decades and has been used as an assurance for exponential grow in the computational and memory capabilities of all computational hardware.

Some of the most widely-used machine learning techniques remain deterministic algorithms such as *K-means clustering* (Section 2.1) and it can be very helpful to look at the relationship between such techniques and discrete latent variable models; in particular how such deterministic algorithms can be derived as restricted inference algorithms for probabilistic models. In the case of K-means, we can understand the general assumptions it places on the data by looking at its relation to the GMM and testing it on synthetic GMM data: K-means implies shared cluster covariance across all clusters (see Figure 1.2); equal density clusters (see Figure 1.1); spherical cluster geometry (see Figure 1.3); known, fixed K and lack of robustness to even trivial outliers (see Figure 1.4). Each of those pitfalls arises from placing certain assumptions on the related underlying GMM. Similar argument can be made for the BNP extension of K-means -- the DP-means algorithm (Kulis & Jordan, 2011) and its relation to the *Dirichlet process mixture model* (Section 2.5.4). In fact K-means, DP-means and many other algorithms can be seen as deterministic algorithms for inference in latent variable models after applying *small variance asymptotic*s (SVA) assumptions(see Section 3.2).

The reason we consider the probabilistic generalization of such techniques is so that we can revisit some of the assumptions we place on probabilistic models in our search for efficient, fast and flexible inference algorithms. We try to rigorously follow the trade-off between flexibility of the inference method and its computational efficiency(in terms of both computational speed and memory requirements). Towards this end, we map this trade-off for some of the most popular inference algorithms in the case of widely-used discrete latent variable models (such as mixture models and *hidden Markov models*) and their BNP extensions. We make an attempt to extend the applications for BNP models by proposing an *iterative MAP* (Maximum a posteriori) framework for flexible deterministic inference which can process large datasets and can operate on resource-constraint hardware, while not changing the structure of the underlying model.

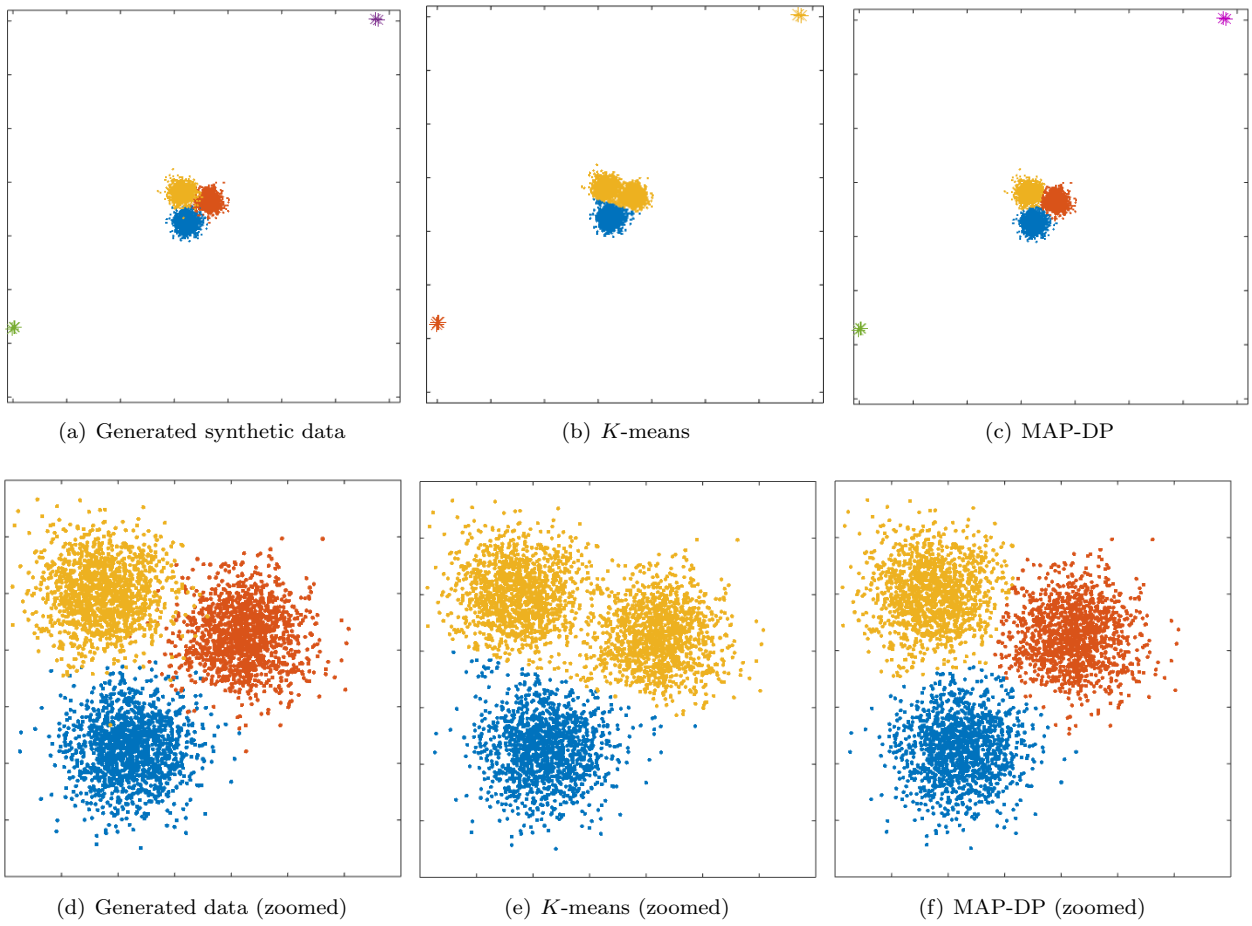
Other ubiquitous methods for efficient inference in BNP models rely on *variational Bayes* (VB) approximations (Blei & Jordan, 2006; Teh *et al.*, 2007; Broderick *et al.*, 2013b; Foti *et al.*, 2014; Hughes & Sudderth, 2013) which we discuss throughout the chapters of this thesis. Typically, VB methods are a lot harder to derive than similar iterative MAP algorithms and require additional assumptions about a model in order to make inference feasible at all.

1.2 Contributions

After more than 50 years, the K-means algorithm remains the preferred clustering tool for most real world applications (Berkhin, 2006). In this thesis we study algorithms such as K-means from a probabilistic vantage point: as a restricted (SVA) case of a latent discrete probabilistic model. From this probabilistic view we can better – and more rigorously – understand the assumptions inherent with widely-used clustering methods and explore how each of those assumptions influences the flexibility, the simplicity and the usefulness of the corresponding clustering method. This sets a general framework for us to derive simple model-based algorithms, which at the cost of just slight departure from existing algorithms inherit greater flexibility and many useful statistical properties. The resulting contributions in this thesis are listed below:

- We derive a modified version of K-means: *collapsed K-means* which is as conceptually simple, but is more robust to changes in initialization of the parameters, and is less likely to converge to a poor local solution than the original K-means. A novel *K-means with reinforcement* algorithm is proposed which overcomes the implicit assumption of K-means that data is shared equally across the K clusters.

- In contrast to methods obtained using SVA assumptions to probability models, we propose an iterative, *greedy* MAP algorithm for deriving model-based deterministic methods which are only marginally more



(a) Generated synthetic data (b) K-means (c) MAP-DP

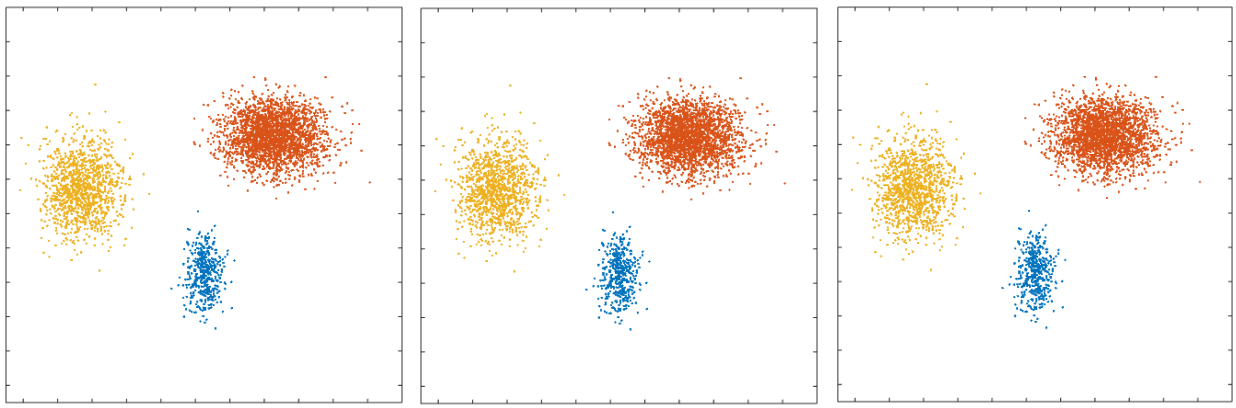
(d) Generated data (zoomed) (e) K-means (zoomed) (f) MAP-DP (zoomed)

Figure 1.4: Clustering performed by K-means and MAP-DP for spherical, synthetic Gaussian data, with outliers. All clusters have the same radii and density. There are two outlier groups with two outliers in each group. K-means fails to find a good solution where MAP-DP succeeds; this is because K-means puts some of the outliers in a separate cluster, thus inappropriately using up one of the $K = 3$ clusters. This happens even if all the clusters are spherical, equal radii and well-separated.

complex than traditional algorithms such as K-means. Some specific applications of greedy MAP have already been studied in different domains (Bertoletti *et al.*, 2015; Besag, 1986). However, here we formalize this framework and study its potential applied to different constructions of popular parametric and BNP discrete latent variable models. We systematically demonstrate the practical and conceptual advantages of iterative MAP compared to more restrictive SVA methods (Broderick *et al.*, 2013a) for inference in BNP models.

- We derive deterministic methods for inference in the *Dirichlet process mixture model* (DPMM): the *MAP-DPMM* and the collapsed *MAP-DPMM* (Raykov *et al.*, 2016c) algorithms which can be used for both approximate inference or simple nonparametric clustering algorithms which learn the number of clusters from the data. We evaluate the MAP-DPMM methods on benchmark and synthetic datasets and further compare them to both standard parametric and nonparametric clustering alternatives. We demonstrate applications of these novel methods for discovering phenotypes of Parkinson's disease from a rich patient dataset and for nonparametric analysis of longitudinal health data.

- We present an intuitive interpretation of the *hierarchical Dirichlet process* (HDP) motivating some new



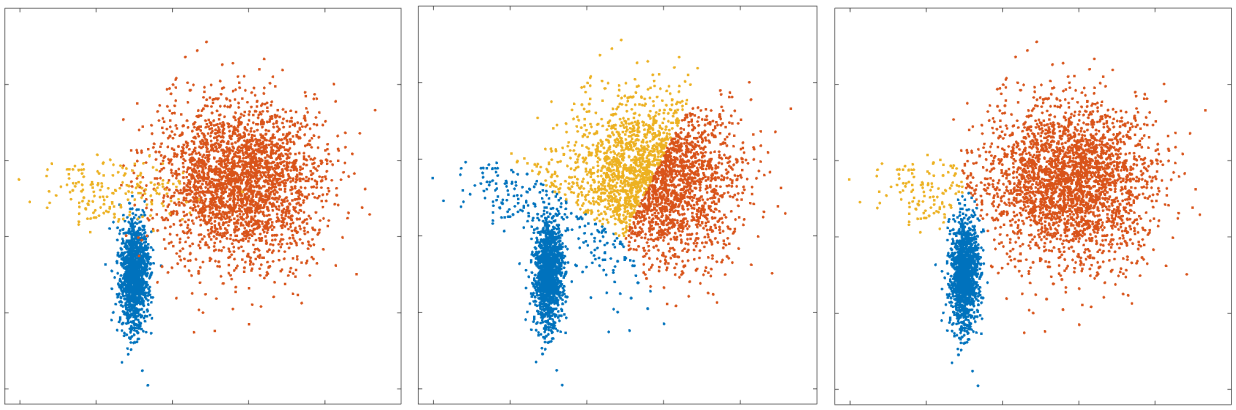
(a) Generated synthetic data (b) K-means (c) MAP-DP

Figure 1.5: Clustering solution obtained by K-means and MAP-DP for synthetic elliptical Gaussian data. The clusters are trivially well-separated, and even though they have different densities (12% of the data is blue, 28% yellow cluster, 60% orange) and elliptical cluster geometries, K-means produces a near-perfect clustering, as with MAP-DP. This shows that K-means can in some instances work when the clusters are not equal radii with shared densities, but only when the clusters are so well-separated that the clustering can be trivially performed by eye.

applications for it as a clustering model for the standard clustering problem of data with mixed continuous and categorical data types. The different constructions of the HDP are conceptually contrasted and a novel deterministic clustering method, *MAP-HDP,* is proposed which outperforms the existing SVA alternative.

- Two novel nonparametric algorithms for analysis of sequential data are proposed which we call *MAP-iHMM* (Raykov *et al.*, 2015a, 2016b) and *dynamic MAP-iHMM.* The dynamic MAP-iHMM can be seen as a nonparametric extension of the classical *Viterbi algorithm* for inference in hidden Markov models (HMMs). We demonstrate the applicability of MAP-iHMM and dynamic MAP-iHMM to some synthetic and real world examples, where MAP methods reach local clustering solutions orders of magnitude faster than current MCMC methods. Applications include a problem in genomic hybridization and an automated quality control for the analysis of accelerometer data collected during a walking test using a smartphone.

- A novel study is performed on the challenging problem of predicting room occupancy head count using a single passive infrared (PIR) sensor (Raykov *et al.*, 2016a). A state-of-the art system is proposed which can provide occupancy estimates every 30 seconds; the estimates are typically within +1/-1 individual of the true number of occupants. We demonstrate how, using MAP-iHMM, the whole system can be sufficiently optimized to allow it to work and segment data directly onto a highly resource-constrained microcontroller board. The loss in accuracy of the system when segmentation is done using MAP-iHMM compared to more expensive MCMC methods is negligible in practice, but this optimization allows the whole system to be deployed as a self-contained product without the need for expensive supporting computational hardware.



(a) Generated synthetic data (b) K-means (c) MAP-DP

Figure 1.6: Clustering solution obtained by K-means and MAP-DP for overlapping, synthetic elliptical Gaussian data. All clusters have different elliptical covariances, and the data is unequally distributed across different clusters (30% blue cluster, 5% yellow cluster, 65% orange). The significant overlap is challenging even for MAP-DP, but it produces a meaningful clustering solution where the only mislabelled points lie in the overlapping region. K-means does not produce a clustering result which is faithful to the actual clustering.

1.3 Thesis organization

The main body if this thesis starts in Chapter 2 with a review of some of the relevant fundamental concepts in probabilistic modeling and pattern recognition related to clustering. The review includes discussion on Gaussian mixture models from both frequentist and Bayesian perspectives; the most relevant methods used for inference in mixture models, as well as some challenges that we face depending on the modeling perspective, construction and inference method. The second part of Chapter 2 reviews the construction and properties of DPs and the DPMM.

In Chapter 3 we start by revisiting the well known connection between K-means and mixture models. This connection motivates the construction of a new version of K-means which is related to collapsed mixture models. Mirroring some of the latest work on SVA, we also derive a novel K-means with reinforcement. In order to relax some of the restrictive assumptions that SVA clustering algorithms impose, we motivate the use of iterative MAP methods. This sets the stage for an in-depth exploration of an entire framework of deterministic methods for inference in DPMMs. We review the most widely-used inference strategies for DPMMs and introduce MAP-DPMM (see Raykov *et al.* and Raykov *et al.*). The practical relevance of the proposed MAP methods is demonstrated on a problem typically attempted using K-means: discovery of phenotypes of Parkinson's disease and parkinsonism. The Chapter concludes with some further applications of MAP-DPMM as a building block for more complex models.

Chapter 4 reviews in the detail the hierarchical DP (HDP), introduced in Teh *et al.* for modeling data that originates in different dependent subsets. We review various constructions and inference methods for HDP mixtures and propose a novel method for *multi-level clustering*, *MAP-HDP*. This is compared against the few existing deterministic algorithms for inference in HDP mixtures and tested against the SVA algorithm. HDPs are also used as a building block for the models which appear later in Chapter 5 for sequential data.

As with earlier chapters, in Chapter 5 we review various MCMC and SVA methods for inference in iHMMs. We introduce a novel MAP method for sequence clustering which takes advantage of dynamic programming and also propose the novel MAP-iHMM method (see Raykov *et al.*) as a slower, but sometimes

more accurate alternative. Where the accuracy of MAP methods is reduced for more complex hierarchical models, we demonstrate visually that in lower dimensional datasets most of the important states can be recovered in just a few iterations.

Chapter 6 follows different structure than earlier chapters in order to introduce the reader to the challenging problem of room occupancy estimation as one of the fundamental tasks of "self-aware environments". It starts with some necessary motivation and review of the relevant work for the problem of predicting room occupancy. We briefly describe our experiments, hypothesis and hardware used for data collection. Once the problem is formulated from a statistical and engineering perspective, we motivate the need for the iHMMs as a part of a rigorous approach to solving the problem. Different iHMM inference algorithms are tested and their effect on the trade-off between accuracy and computational efficiency is assessed. We demonstrate that using MAP-iHMM, we can deploy a practically useful, self-contained system that can perform all of its computation and inference on a cheap microcontroller board with limited computational hardware resources (see Raykov *et al.*)[3].

The final Chapter of this thesis draws some general conclusions and proposes directions for future work.

[3]A patent application has been submitted on behalf of the company ARM to the US patent office disclosing this system. The patent is named: "Predicting the number of occupants with a single PIR sensor using behaviour extraction".

Chapter 2

Discrete latent variable models and inference

Mixture models are widely-used discrete latent variable models most often selected for their ability to represent inherent sub-groups and identify clusters in a rigorous way. This chapter starts by first reviewing the nonprobabilistic K-means clustering algorithm and its pitfalls. Then it proceeds with discussion of the Gaussian mixture model (GMM) which can be used to cluster data overcoming a lot of the drawbacks of K-means. We extend this exposition to include the Bayesian setting of mixture models as well as some fundamental principles for performing inference in Bayesian probabilistic models. Many of the concepts reviewed and presented in this chapter serve as foundation for deriving more complicated models and inference methods later on. The second part of the chapter focuses on reviewing the definitions, properties and the various constructions of the Dirichlet process (DP). The DP will serve as a building block for most of the flexible nonparametric probabilistic models we discuss in later chapters. We conclude the chapter with a short overview of the main inference algorithms discussed in this thesis and the associations between them.

2.1 The K-means algorithm

K-means was first introduced as a method for *vector quantization* in communication technology applications (Lloyd, 1982), yet it is still one of the most widely-used clustering algorithms. For example, in discovering *clinical sub-types* of Parkinson's disease, we observe that most studies have used the K-means algorithm to find sub-types in patient data (van Rooden *et al.*, 2010). It is also the method of choice in *visual bag of words* models in automated image understanding (Fei-Fei & Perona, 2005). Perhaps the major reasons for the popularity of K-means are *conceptual simplicity* and *computational scalability*, in contrast to more flexible clustering methods.

For the ensuing discussion, we will use the following mathematical notation: let us denote the data as $X = (x_1, \ldots, x_N)$ where each of the N data points x_i is a D-dimensional vector; denote the *cluster assignment* associated to each data point by $z_1, \ldots, z_N$, where if data point x_i belongs to cluster k we write $z_i = k$. The parameter $\epsilon > 0$ is a small threshold value to assess when the algorithm has converged on a good solution and should be stopped (typically $\epsilon = 10^{-6}$). Using this notation, K-means can be written as in Algorithm **2.1**.

To paraphrase this algorithm: it alternates between updating the assignments of data points to clusters while holding the estimated cluster *centroids*, μ_k, fixed, and updating the cluster centroids while holding the

assignments fixed. It can be shown to find *some* minimum (not necessarily the *global*, i.e. smallest of all possible minima) of the following *objective function*:

$$E = \frac{1}{2} \sum_{k=1}^{K} \sum_{i:z_i=k} \|x_i - \mu_k\|_2^2 \tag{2.1}$$

with respect to the set of all cluster assignments z and cluster centroids μ, where $\frac{1}{2} \|.\|_2^2$ denotes the (square of the) *Euclidean distance* (distance measured as the sum of the square of differences of coordinates in each direction). In fact, the value of E *cannot increase* on each iteration, so, eventually E will stop changing and K-means will converge.

	Algorithm 2.1: K-means		**Algorithm 2.2**: MAP-GMM (spherical Gaussian)
Input	$x_1, \ldots, x_N$: D-dimensional data		$x_1, \ldots, x_N$: D-dimensional data
	$\epsilon > 0$: convergence threshold		$\epsilon > 0$: convergence threshold
	K: number of clusters		α: concentration parameter
			σ^2: spherical cluster variance
			σ_0^2: prior centroid variance
Output	$z_1, \ldots, z_N$: cluster assignments		$z_1, \ldots, z_N$: cluster assignments
	$\mu_1, \ldots, \mu_K$: cluster centroids		$\mu_1, \ldots, \mu_K$: cluster centroids
			$\pi_1, \ldots, \pi_K$: cluster weights
1	Set μ_k for all $k \in 1, \ldots, K$	**1**	Set μ_k and π_k for all $k \in 1, \ldots, K$
2	$E_{\text{new}} = \infty$	**2**	$E_{\text{new}} = \infty$
3	**repeat**	**3**	**repeat**
4	$E_{\text{old}} = E_{\text{new}}$	**4**	$E_{\text{old}} = E_{\text{new}}$
5	**for** $i \in 1, \ldots, N$	**5**	**for** $i \in 1, \ldots, N$
6	**for** $k \in 1, \ldots, K$	**6**	**for** $k \in 1, \ldots, K$
7	$d_{i,k} = \frac{1}{2} \|x_i - \mu_k\|_2^2$	**7**	$d_{i,k} = \frac{1}{2\sigma^2} \|x_i - \mu_k\|_2^2 + \frac{D}{2} \ln \sigma^2 - \ln \pi_k$
8	$z_i = \arg\min_{k \in 1,\ldots,K} d_{i,k}$	**8**	$z_i = \arg\min_{k \in 1,\ldots,K+1} d_{i,k}$
9	**for** $k \in 1, \ldots, K$	**9**	**for** $k \in 1, \ldots, K$
10	$\mu_k = \frac{1}{N_k} \sum_{j:z_j=k} x_j$	**10**	$\mu_k = \frac{\sigma^2 \mu_0 + \sigma_0 \sum_{j:z_j=k} x_j}{\sigma^2 + \sigma_0^2 N_k}$ $\pi_k = \frac{N_k + \alpha/K - 1}{N + \alpha - K}$
11	$E_{\text{new}} = \sum_{k=1}^{K} \sum_{i:z_i=k} d_{i,k}$	**11**	$E_{\text{new}} = \sum_{k=1}^{K} \sum_{i:z_i=k} d_{i,k}$ $- \log \Gamma(N + \alpha) - \sum_{k=1}^{K} \log \Gamma(N_k + \alpha/K)$
12	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$	**12**	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$

Perhaps unsurprisingly, the simplicity and computational scalability of K-means comes at a high cost. In particular, the algorithm is based on quite restrictive assumptions about the data, often leading to severe limitations in accuracy and interpretability:

1. By use of the Euclidean distance K-means treats the data space as *isotropic* (distances unchanged by translations and rotations). This means that data points in each cluster are modeled as lying within

a *sphere* around the cluster centroid. A sphere has the same radius in each dimension. Furthermore, as clusters are modeled only by the position of their centroids, K-means implicitly assumes all clusters have the same radius. When this implicit equal-radius, spherical assumption is violated, K-means can behave in a non-intuitive way, even when clusters are very clearly identifiable by eye (see Figures 1.2, 1.3).

2. The Euclidean distance entails that the average of the coordinates of data points in a cluster is the centroid of that cluster. Euclidean space is *linear* which implies that small changes in the data result in proportionately small changes to the position of the cluster centroid. This is problematic when there are *outliers*, that is, points which are unusually far away from the cluster centroid by comparison to the rest of the points in that cluster. Such outliers can dramatically impair the results of K-means (see Figure 1.4).

3. K-means clusters data points purely on their (Euclidean) *geometric closeness* to the cluster centroid (algorithm line 9). Therefore, it does not take into account the different *densities* of each cluster. So, because K-means implicitly assumes each cluster occupies the same volume in data space, each cluster must contain the same number of data points. We will show later that even when all other implicit geometric assumptions of K-means are satisfied, it will fail to learn a correct, or even meaningful, clustering when there are significant differences in cluster density (see Figure 1.1).

4. The number K of groupings in the data is fixed and assumed known; this is rarely the case in practice. Thus, K-means is quite inflexible and degrades badly when the assumptions upon which it is based are even mildly violated by e.g. a tiny number of outliers (see Figure 1.4).

Some of the above limitations of K-means have been addressed in the literature. Regarding outliers, variations of K-means have been proposed that use more "robust" estimates for the cluster centroids. For example, the *K-medoids* algorithm uses the point in each cluster which is most centrally located. By contrast, in *K-medians* the median of coordinates of all data points in a cluster is the centroid. However, both approaches are far more computationally costly than K-means. K-medoids, requires computation of a pairwise similarity matrix between data points which can be prohibitively expensive for large data sets. In K-medians, the coordinates of cluster data points in each dimension need to be sorted, which takes much more effort than computing the mean. Alternatively, by using the *Mahalanobis distance*, K-means can be adapted to non-spherical clusters (Sung & Poggio, 1998), but this approach will encounter problematic computational singularities when a cluster has only one data point assigned.

Banerjee *et al.* makes use of *Bregman divergence* to unify some of the centroid-based parametric clustering approaches (such as standard K-means evaluated using Euclidean distance and modified K-means evaluated using Mahalanobis distance) as special cases of a more general formulation. The Bregman divergence between any two vectors x and θ is defined as $D_\phi(x,\theta) = \phi(x) - \phi(\theta) - \langle x-\theta, \nabla\phi(\theta)\rangle$ for a differentiable and strictly convex function $\phi : S \to \mathbb{R}$ on a closed convex set $S \subseteq \mathbb{R}^D$, with $\langle . \rangle$ denoting dot product and $\nabla\phi(\theta)$ denoting the gradient vector of ϕ evaluated at θ. Then the K-means objective function can be generalized to:

$$E = \frac{1}{2}\sum_{k=1}^{K}\sum_{i:z_i=k} D_\phi(x_i, \tilde{\mu}_k) \tag{2.2}$$

where $\tilde{\mu}_k = \nabla\phi(\cdot)$ here denotes the expectation parameter of points in cluster k. This more general algorithm does not restrict the data space to be Euclidean, extending to any data space that can be described with

Bregman divergence as a measure of distance[1]. Depending on the data we are dealing with and the geometrical properties we wish to explore, we can specify an appropriate function ϕ.

For example, the square Euclidean distance of K-means can be obtained by choosing $\phi(x) = \langle x, x \rangle$. The chosen underlying function is strictly convex and differentiable on $\mathbb{R}^D$ and writing the above definition of Bregman divergence we get:

$$\begin{aligned} D_\phi(x, \theta) =& \langle x, x \rangle - \langle \theta, \theta \rangle - \langle x - \theta, \nabla\phi(\theta) \rangle \\ =& \langle x, x \rangle - \langle \theta, \theta \rangle - \langle x - \theta, 2\theta \rangle \\ =& \langle x - \theta, x - \theta \rangle = \|x - \theta\|_2^2 \end{aligned} \tag{2.3}$$

Alternatively, if we choose $\phi(x) = x^T A x$ for A being the inverse of the covariance matrix we can express the Mahalanobis distance as Bregman divergence:

$$\begin{aligned} D_\phi(x, \theta) =& x^T A x - \theta^T A \theta - \langle x - \theta, \nabla\phi(\theta) \rangle \\ =& x^T A x - \theta^T A \theta - \langle x - \theta, 2\theta A \rangle \\ =& x^T A x + \theta^T A \theta - 2x^T A \theta = (x - \theta)^T A (x - \theta) \end{aligned} \tag{2.4}$$

therefore the non-spherical variant of K-means from (Sung & Poggio, 1998) can be also seen as a special case of the general Bregman divergence clustering algorithm optimizing the objective in (2.2). The clustering problems which we can express using the objective function (2.2) have the useful property that a simple approximate procedure exists that optimizes the corresponding objective. Furthermore, the Bregman divergence representation is often useful due to its relationship to the *exponential family* of distributions. Every *regular exponential family distribution*[2] is associated with a unique Bregman divergence in the following way: the log-likelihood of the density of an exponential family distribution can be written as the sum of the negative of a uniquely determined Bregman divergence and a function that does not depend on the distribution parameters Forster & Warmuth (2002). This defines an important association between the exponential family distributions and associated Bregman divergences. Somewhat more sophisticated procedures such as K-medoids cannot necessarily be included in this framework.

Clustering with some K-means alternatives that exploit different distance measures may adequately address issues such as non-spherical data (Issue 1) and outliers (Issue 2). However, all algorithms derived to optimize an objective function of the form of (2.2) will cluster data purely based on its geometric closeness (Issue 3) and will require fixing K in advance (Issue 4). In addressing the problem of the fixed number of clusters K, note that it is not possible to choose K simply by clustering with a range of values of K and choosing the one which minimizes E. This is because K-means is *nested*: we can always decrease E by increasing K, even when the true number of clusters is much smaller than K, since, all other things being equal, K-means tries to create an equal-volume partition of the data space. Therefore, data points find themselves ever closer to a cluster centroid as K increases. In the extreme case for $K = N$ (the number of data points), then K-means will assign each data point to its own separate cluster and $E = 0$, which obviously has no meaning as a "clustering" of the data.

[1] Bregman divergence is similar to a metric, but does not satisfy the triangle inequality nor symmetry.

[2] Distributions from the regular exponential family are exponential family distributions with parameter space being an open set, i.e. all of the parameters from the parameter space, $\theta \in \Theta$, and lie on the interior, $\Theta \equiv \text{interior}(\Theta)$.

2.2 Mixture models

While K-means essentially takes into account only the geometry of the data, mixture models are inherently *probabilistic*, that is, they involve fitting a probability density model to the data. The advantage of considering this probabilistic framework is that it provides a *mathematically principled* way to understand the algorithm's limitations and assumptions, while introducing further flexibility. We assume that the data can be generated using a probability density of certain form (in this case mixture of Gaussian distributions) and we seek to learn the best parametrization of that form (see Figure 2.1(b)). Estimating a mixture density model for the data is a more general problem than the clustering one, as here we do not assume every point necessarily belongs to one of the underlying clusters. Instead, each observation has a non-zero probability of belonging to each of the K clusters.

In Gaussian mixture models (Bishop, 2006, page 430) we assume that data points are drawn from a *mixture* (a weighted sum) of Gaussian distributions with density $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \,|\mu_k, \Sigma_k)$, where K is the fixed number of components, $\pi_k > 0$ are the weighting coefficients with $\sum_{k=1}^{K} \pi_k = 1$, and μ_k, Σ_k are the parameters of each Gaussian in the mixture. So, to produce a data point x_i, the model first draws a cluster assignment $z_i = k$. The distribution over each z_i is known as a *categorical distribution* with K parameters $\pi_k = p(z_i = k)$. Then, given this assignment, the data point is drawn from a Gaussian with mean μ_{z_i} and covariance Σ_{z_i}.

Under this model, the conditional probability of each data point given its cluster assignment is $p(x_i \,|z_i = k) = \mathcal{N}(x_i \,|\mu_k, \Sigma_k)$, which is just a Gaussian. But an equally important quantity is the probability we get by reversing this conditioning: the probability of an assignment z_i given a data point x_i (sometimes called the *responsibility*), $p(z_i = k \,|x_i)$. This raises an important point: in the GMM, a data point has a finite probability of belonging to *every* cluster, whereas, for K-means each point belongs to only one cluster. This is because the GMM is *not* a partition of the data: the assignments z_i are treated as random draws from a distribution.

One of the most widely-used algorithms for estimating the unknowns of a GMM from some data (that is the variables z, μ, Σ and π) is the *Expectation-Maximization* (E-M) algorithm. This iterative procedure alternates between the E (*expectation*) step and the M (*maximization*) steps. The E-step uses the responsibilities to compute the cluster assignments, holding the cluster parameters fixed. The M-step re-computes the cluster parameters holding the cluster assignments fixed:

E-step: Given the current estimates for the cluster parameters, compute the responsibilities:

$$\gamma_{i,k} = p(z_i = k \,|x_i, \pi, \mu, \Sigma) = \frac{\pi_k \mathcal{N}(x_i \,|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \,|\mu_j, \Sigma_j)} \tag{2.5}$$

M-step: Compute the parameters that maximize the *likelihood* of the data set $p(X \,|\pi, \mu, \Sigma)$, which is the probability of all of the data under the GMM (Dempster *et al.*, 1977):

$$p(X \,|\pi, \mu, \Sigma) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \,|\mu_k, \Sigma_k) \tag{2.6}$$

Maximizing this with respect to each of the parameters can be done in closed form:

$$\begin{array}{cc} S_k = \sum_{i=1}^{N} \gamma_{i,k} & \pi_k = \frac{S_k}{N} \\ \mu_k = \frac{1}{S_k} \sum_{i=1}^{N} \gamma_{i,k} x_i & \Sigma_k = \frac{1}{S_k} \sum_{i=1}^{N} \gamma_{i,k} (x_i - \mu_k)(x_i - \mu_k)^T \end{array} \tag{2.7}$$

Each E-M iteration is guaranteed not to decrease the likelihood function $p(X \,|\pi, \mu, \Sigma)$. So, as with K-

means, convergence is guaranteed, but not necessarily to the global maximum of the likelihood. We can, alternatively, say that the E-M algorithm attempts to minimize the GMM objective function:

$$E = -\sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x_i \,|\mu_k, \Sigma_k\right) \tag{2.8}$$

When changes in the likelihood are sufficiently small the iteration is stopped. If used as a clustering tool, E-M for GMM definitely adds to the computational and conceptual complexity of K-means, but resolves some of the issues discussed earlier (the issues of inherent sphericity 1 and purely geometry based clustering 3 from Section 2.1). At the same time, even when assuming that the observed data is generated from a mixture of Gaussians, there are certain issues with the E-M algorithm for GMM to keep in mind:

1. The convergence of E-M is guaranteed only to a local solution and typically finding the globally optimal parameters of objective in (2.8) will not be feasible. The quality of this local solution depends upon careful initialization.

2. The probabilistic nature of the GMM allows us to incorporate uncertainty in the clusters we learn from the data by estimating probabilities for each assignment variable, rather then just learning some specific cluster assignment values. However, the uncertainty in the component parameters, π, μ and Σ, is not explicitly modeled which leads to sensitivity of the model to initialization and poorer performance in the presence of outliers and potential for overfitting .

3. The minimization of the GMM objective function (2.8) using E-M algorithm can be often lead to a singularity in the following way: if one of the Gaussian components 'collapses' onto a specific data point (meaning $\mu_k \to x_i$ and $\Sigma_k \to 0$), the objective goes to (minus) infinity, $E \to -\infty$. These singularities of the GMM are considered as an example of the severe overfitting that sometimes occurs in maximum likelihood methods.

4. The number of Gaussian components K describing the data is assumed fixed and known. Furthermore, the likelihood $p\left(X \,|\pi, \mu, \Sigma\right)$ does not allow for adequate model selection for various K, as it will always tolerate larger K until components start collapsing on single points.

2.3 The Bayesian framework

2.3.1 Bayesian mixture models

A natural way to address many of the issues with GMMs (such as: getting 'stuck' at local optima; potential overfitting; 'point' estimates of the parameters π, μ and Σ), is to incorporate an additional level of hierarchy in the *probabilistic graphical model* (PGM) therebyadopting the Bayesian modelling framework. Under the Bayesian paradigm, we specify a prior distribution over each of the unknown model parameters in a PGM. This allows us to express a posterior distribution over each of the model parameters which incorporates both information gained from the data and information gained from the prior. By contrast, in Section 2.2 we were computing only point estimates for the parameters π, μ and Σ. Typically, for each of the prior distributions we will need to specify some new, corresponding *hyperparameters*. The values of these parameters can be either specified to reflect some additional (expert) knowledge about the corresponding quantity, or using other approaches, some of which we discuss in Appendix A. In practice we often choose the prior distributions over the parameters in the model to be *conjugate* to the parameter likelihood. Conjugacy between the prior and

the likelihood for a random variable guarantees the same mathematical form for the prior and posterior and simplifies the mathematics.

Gaussian mixtures

Let us first consider a Bayesian treatment of the GMM from Section 2.2. The conjugate prior over the categorically distributed mixing coefficients $(\pi_1, \ldots, \pi_K)$ is the Dirichlet, where in the absence of additional information it is typically assumed uniform, or $\pi \sim \text{Dir}(\alpha/K, \ldots, \alpha/K)$ for some *concentration parameter* $\alpha > 0$. If we assume the cluster parameters of the Gaussian components are unknown, one quite general approach is to use a *Normal-Inverse-Wishart* (NIW) over the joint (μ_k, Σ_k) for $k = 1, \ldots, K$. We can then write a probabilistic model for generating data (*generative model*) from this Bayesian GMM:

$$\begin{aligned} (\mu_k, \Sigma_k) &\sim \text{NIW}(m_0, c_0, b_0, a_0) \\ \pi &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \\ z_i &\sim \text{Categorical}(\pi) \\ x_i &\sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \end{aligned} \tag{2.9}$$

for $k = 1, \ldots, K$ and $i = 1, \ldots, N$ with $'X \sim F'$ denoting that random variable X has distribution F. We denote the NIW prior hyperparameters with (m_0, c_0, b_0, a_0) where the vector m_0 reflects our prior belief for the means of the cluster components; the positive scalar c_0 controls the scale between the covariance in the Gaussian prior over the cluster means and the covariance matrix drawn from an *Inverse-Wishart* prior; b_0 is the *inverse scale* matrix and a_0 is a positive scalar parameter denoting the *degrees of freedom* of the Inverse-Wishart prior.

In this Bayesian formalism, it is straightforward to construct simpler models that assume fewer unknown parameters. For example, if we believe that the Gaussian components describing each cluster are approximately spherical, it can be efficient to assume $\Sigma_k = \sigma_k \mathbf{I}$ ($\mathbf{I}$ denoting the identity matrix with same dimension as the data) for $k = 1, \ldots, K$ and place a simpler *Normal-Inverse-Gamma* prior over the parameters, $(\mu_k, \sigma_k) \sim \text{NIG}(m_0, c_0, b_0, a_0)$. Alternatively, often we assume that the covariance matrices are known to simplify the computation, then we place a simple Gaussian prior over only the cluster means, $\mu_k \sim \mathcal{N}(\mu_0, \sigma_0)$ for $k = 1, \ldots, K$.

Exponential family mixtures

The notion of mixture models (Bayesian or not) is not constrained to only Gaussian data and under the same framework, we can model a large range of data types. In fact, the only major restriction we place on the mixtures and other probability models discussed in this thesis is the existence of conjugate priors for each of the model terms. A common and quite flexible family of such conjugate models is to write them in more general *exponential family* mixture model form. This is useful because any exponential family distribution is guaranteed to have another exponential family conjugate prior distribution available in closed form (vice versa is not always the case). The Gaussian distribution is just one of the exponential family distributions, therefore we can view the GMM as a special case of the exponential family mixture model:

$$\begin{aligned} \theta_k &\sim G_0 \\ \pi &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \\ z_i &\sim \text{Categorical}(\pi) \\ x_i &\sim F(\theta_{z_i}) \end{aligned} \tag{2.10}$$

where $\theta_1, \ldots, \theta_K$ are the component parameters; $\pi_1, \ldots, \pi_K$ are the mixing parameters; F is an exponential family distribution and G_0 is conjugate to F. Given a data point i is associated with component indicated by the value of z_i, the probability density function of $F(\theta_{z_i})$ is written in the form:

$$p(x_i|\theta_{z_i}) = \exp(\langle g(x_i), \theta_{z_i}\rangle - \psi(\theta_{z_i}) - h(x_i)) \tag{2.11}$$

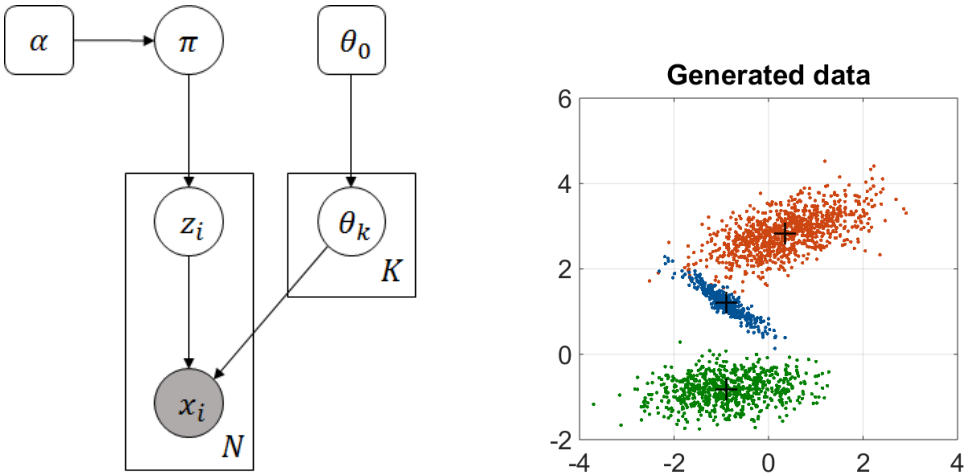
where $g(.)$ is the *sufficient statistic* function, $\psi(\theta_{z_i}) = \log \int \exp(\langle x_i, \theta_{z_i}\rangle - h(x_i))\, dx_i$ is the *log partition* function and $h(x_i)$ the *base measure* of the distribution. As the prior over the component parameters G_0 is conjugate to F, we can obtain its probability density function as well in closed form:

$$p(\theta|\tau, \eta) = \exp(\langle \theta, \tau\rangle - \eta\psi(\theta) - \psi_0(\tau, \eta)) \tag{2.12}$$

where (τ, η) are the prior hyperparameters of the prior measure G_0. From Bayesian conjugacy, the posterior $p(\theta_k|x, \tau_k, \eta_k)$ will take the same form as the prior where the prior hyperparameters τ and η will be updated to $\tau_k = \tau + \sum_{j:z_j=k} g(x_j)$ and $\eta_k = \eta + N_k$ with $N_k = \sum_{j:z_j=k} 1$.

For example, in the specific case of a GMM with unknown means and covariances, we replace in (2.10) F with the Gaussian distribution; G_0 with the Normal-Inverse-Wishart; component parameters θ with (μ, Σ); the hyperparameters (τ, η) with (m, c, b, a) and we can recover the model from (2.9). Examples of other mixture models can be obtained by substituting the relevant expressions from Appendix A.

The Bayesian mixture model can be seen as a more general treatment to mixture modeling as it allows for more control over the random parameters and it allows for more principled treatment of the model uncertainty. In the Bayesian GMM for example the model parameters no longer depend only on the data, but rather reflect a balanced trade-off between our belief about them expressed through (m_0, c_0, b_0, a_0) and the data X we have observed. Furthermore, Bayes rule provides us with principles to integrate out any *nuisance* model parameters which are not of explicit interest in the particular problem. This will allow us to vary the structure of the model, which can potentially be used for: more efficient inference, better parameter initialization, and better prediction and/or model selection. In fact, according to the Bayesian modeling paradigm, placing priors over the unknown quantities in the model and integrating over them is always the "correct" thing to do, unless sufficient information is available to fix the parameters to some particular values.



(a) Graphical model

(b) GMM generated data

Figure 2.1: Probabilistic graphical model of the Bayesian mixture model. In the Gaussian case $\theta = (\mu, \Sigma)$ and $\theta_0 = (m_0, c_0, b_0, a_0)$.

2.3.2 Gibbs sampling

Where the E-M algorithm is the usual choice for inference in the GMM from Section 2.2, inference in complex Bayesian models is usually performed using *Markov chain Monte Carlo* (MCMC) methods. Recall that the E-M algorithm is a maximum likelihood approach and so only guaranteed to find locally optimal fit of the model to the data. By contrast, the Gibbs sampler, introduced in (Geman & Geman, 1984), is a randomized algorithm and as with all MCMC methods is asymptotically (that is, after an infinite number of iterations) guaranteed to find the global posterior distribution of the model. Unfortunately these asymptotic guarantees are not very useful in practice, as we never have unconstrained computational resources at our disposal and the Gibbs sampler can take a prohibitively large number of iterations to converge to the posterior distribution. This is worsened by *poor mixing* of the sampler when the required posterior consists of few "islands" of states with high probability surrounded by an "ocean" of small non-zero probability. Gibbs sampling can also be seen as a specific case of the *Metropolis-Hasting* (M-H) algorithm, which in its varying forms can better handle discontinuities in the posterior state space. This thesis does not explore M-H in detail, but we would direct the reader to (Chib & Greenberg, 1995) for an intuitive presentation.

Each step of the Gibbs sampler involves replacing the value of one of the variables in the model by a value drawn from the distribution of that variable conditioned on the values of the rest of the variables in the model. For the Bayesian GMM (2.9), the variables (parameters and latent variables) would be $\{z_1, ...z_N, \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K, \pi_1, ..., \pi_K\}$. Gibbs iterations would involve sampling the mixture component parameters $\mu_1, ..., \mu_K$ and $\Sigma_1, ..., \Sigma_K$; the mixture coefficients $\pi_1, ..., \pi_K$ and the cluster indicators $z_1, ..., z_K$ given the data $x_1, ..., x_N$. At each iteration, holding the rest of the quantities fixed, we will update each z_i by drawing samples from the categorical distribution defined with weights for each category k being:

$$p\left(z_i = k \,|\mu_k, \Sigma_k, \pi_k, x\right) = \frac{\pi_k \mathcal{N}\left(x_i \,|\mu_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(x_i \,|\mu_j, \Sigma_j\right)} \tag{2.13}$$

for $k = 1, ..., K$. Conditioned on the parameters $\{\mu, \mathbf{\Sigma}, \pi\}$, the probability of component assignments is computed in the same way as in (2.5). Once the indicator variables have been updated, we proceed by drawing samples now for the component parameters holding the rest of the quantities fixed:

$$\left(\mu_k, \Sigma_k\right) \sim \text{NIW}\left(\mu, \Sigma \,|m_k, c_k, a_k, b_k\right) \tag{2.14}$$

The parameters (m_k, c_k, b_k, a_k) of this NIW distribution depend upon the current values of the indicators and are updated using:

$$\begin{aligned} m_k &= \tfrac{c_0 m_0 + N_k \bar{x}_k}{c_0 + N_k} \\ c_k &= c_0 + N_k \\ a_k &= a_0 + N_k \\ b_k &= b_0 + S + \tfrac{c_0 N_k}{c_0 + N_k} \left(\bar{x}_k - m_0\right) \left(\bar{x}_k - m_0\right)^T \end{aligned} \tag{2.15}$$

where $\bar{x}_k = \frac{\sum_{i:z_i=k} x_i}{N_k}$; N_k denotes the number of observations assigned to cluster k and $S = \sum_{i=1}^{K} \left(x_i - \bar{x}_k\right) \left(x_i - \bar{x}_k\right)^T$ is the sample covariance matrix. Note that while (m_0, c_0, b_0, a_0) denote the prior terms of the NIW distribution and should be specified a priori, (m_k, c_k, b_k, a_k) are the corresponding posterior terms of the NIW posterior estimated using the data and the information about the prior. We emphasize the fact that the values of the parameters (m_k, c_k, b_k, a_k) are different for each cluster.

The next step is to sample the mixing coefficients from the following (posterior) Dirichlet distribution:

$$\pi \sim \text{Dir}\left(N_1 + {}^{\alpha}/_{K}, ..., N_K + {}^{\alpha}/_{K}\right) \tag{2.16}$$

The posterior over the mixture weights is a Dirichlet distribution and so keeps the form of its conjugate prior, as expected. The algorithm iterates between sampling each of the random quantities until convergence, however note that convergence in MCMC methods has a different meaning. In the case of E-M for GMM, the complete data likelihood (or equivalently the negative log likelihood (NLL) $-\ln\left(p\left(x,z|\mu,\Sigma,\pi\right)\right)$ in (2.17)) eventually stops increasing (choosing a small threshold value for these changes in likelihood suffices to stop the algorithm when the solution is sufficiently accurate). In Gibbs sampling though the likelihood never converges onto single point solution as it is stochastic. Instead, Gibbs sampling converges onto the required (by design) stationary posterior distribution. Detecting this type of convergence is a complex, well studied and yet still unresolved problem. There are a plethora of possible convergence diagnostics, but none of them provide any theoretical guarantees. Most of them rely on computing at each iteration the complete data likelihood:

$$p\left(x,z|\mu,\Sigma,\pi\right)=\prod_{i=1}^{N}\prod_{k=1}^{K}\pi_{k}^{\delta_{z_i,k}}p\left(x_i|\mu_k,\Sigma_k\right)^{\delta_{z_i,k}} \tag{2.17}$$

where $\delta_{z_i,k}$ is the Kronecker delta. Then, at convergence the sequence of values of $p\left(x,z|\mu,\Sigma,\pi\right)$ estimated for consecutive iterations of the sampler should be independent; there should be no correlation between successive draws of the Gibbs sampler. In practice this can be quite hard to assess as it involves executing many iterations of the sampler ahead to check for correlations. In this thesis we rely on one of the most widely-used convergence diagnostics for Gibbs sampling described in (Raftery & Lewis, 1992). We provide a short outline of Gibbs sampling for the special case of inference in the spherical Bayesian GMM and this can be found in Chapter 3, Algorithm 3.3.

In the more general exponential family formulation of the mixture model (2.10), the Gibbs sampler iterates between updates for the indicators $z_1,\ldots,z_N$, the parameters $\theta_1,\ldots,\theta_K$ and the mixing parameters $\pi_1,\ldots,\pi_K$. Each z_i is updated by sampling from the categorical distribution with weights:

$$p\left(z_i=k\,|\theta_k,\pi_k,x\right)=\frac{\pi_k\exp\left(\left\langle g\left(x_i\right),\theta_k\right\rangle-\psi\left(\theta_k\right)-h\left(x_i\right)\right)}{\sum_{j}^{K}\pi_j\exp\left(\left\langle g\left(x_i\right),\theta_j\right\rangle-\psi\left(\theta_j\right)-h\left(x_i\right)\right)} \tag{2.18}$$

for each component $k=1,\ldots,K$. The component parameters for each k are sampled from the posterior:

$$\theta_k\sim G_0\left(\tau_k,\eta_k\right) \tag{2.19}$$

with $\tau_k=\tau+\sum_{j:z_j=k}g\left(x_j\right)$ and $\eta_k=\eta+N_k$. The mixture coefficients π in the more general setup are still updated from the corresponding Dirichlet posterior from (2.16).

2.3.3 Variational Bayes inference

Variational methods provide a ubiquitous and general framework to convert the problem of stochastic inference to one of deterministic optimization. They have played a key role across many application domains, however here we will briefly review them in the context of probabilistic modeling with the example of the GMM. When doing inference in probabilistic models, we are most often interested in the posterior over all the unknown variables $p\left(Z\,|x\right)$. More precisely, often we try to evaluate the expectation of the complete data log-likelihood (the model log-likelihood) with respect to this posterior. However, this marginalization is rarely tractable, especially in Bayesian models where we have placed a prior distribution over the unknown quantities in the model (the parameters). We already presented one way to approximate stochastically this

expectation using MCMC methods (Gibbs sampler Section 2.3.2); VB methods are deterministic. Consider the distribution $q(Z)$ which approximates the posterior $p(Z|x)$, the log marginal probability of the data can then be written as:

$$\ln p(x) = L(q) + \mathrm{KL}(q\|p) \tag{2.20}$$

where we use:

$$\begin{aligned} L(q) &= \int q(Z) \ln \left\{ \frac{p(x,Z)}{q(Z)} \right\} dZ \\ \mathrm{KL}(q\|p) &= -\int q(Z) \ln \left\{ \frac{p(Z|x)}{q(Z)} \right\} dZ \end{aligned} \tag{2.21}$$

The quantity $L(q)$ can be seen as the lower bound of the posterior $p(Z|x)$, while $\mathrm{KL}(q\|p)$ is the Kullback-Liebler divergence between the approximate $q(Z)$ and the true posterior $p(Z|x)$. In VB inference we aim to maximize the lower bound $L(q)$, or equivalently minimize $\mathrm{KL}(q\|p)$, which implies optimization with respect to $q(Z)$. If no restrictions are placed on the type of distributions $q(Z)$, the maximum of the lower bound is obtained when the KL divergence vanishes since $q(Z) \equiv p(Z|x)$. As this scenario is not tractable, typically restrictions are placed on the family of distributions $q(Z)$ and the problem becomes one of finding the member of the restricted family with minimal KL divergence to the posterior of the model. The choice of restrictions can be somewhat arbitrary, but probably the most common one for inference in Bayesian models is a *factorization* assumption, motivated by *mean field theory* (Parisi, 1988) in physics. That is, we assume that the q distributions factorize to some M disjoint groups, $q(Z) = \prod_{m=1}^{M} q_m(Z_m)$.

Let us present a particular example, the Bayesian GMM from Section 2.3, where data was modeled with:

$$\begin{aligned} (\mu_k, \Sigma_k) &\sim \mathrm{NIW}(m_0, c_0, b_0, a_0) \\ \pi &\sim \mathrm{Dir}(\alpha/K, \ldots, \alpha/K) \\ z_i &\sim \mathrm{Categorical}(\pi) \\ x_i &\sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \end{aligned} \tag{2.22}$$

In this setup $Z = \{z_1, \ldots, z_N, \pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K\}$, the typical factorization assumption is that the joint probability over the latent variables and the component parameters factorizes. We assume $q(z, \pi, \mu, \Sigma) = q(z)\, q(\pi, \mu, \Sigma)$ and the corresponding variational distributions take the form: $q(z) = \prod_{i=1}^{N} \prod_{k=1}^{K} \gamma_{ik}^{\delta_{z_i,k}}$ for γ_{ik} being the probability of point i belonging to component k; $q(\pi) = \mathrm{Dir}(\pi|\alpha)$ with K-dimensional parameter $\alpha = \alpha_1, \ldots, \alpha_K$ and $q(\mu_k, \Sigma_k) = \mathcal{N}\left(\mu_k | m_k, c_k^{-1}\Sigma_k\right) \mathcal{W}^{-1}(\Sigma_k | b_k, a_k)$ denoting the Normal-Inverse-Wishart distribution with:

$$\begin{aligned} m_k &= \tfrac{c_0 m_0 + N_k \bar{x}_k}{c_0 + N_k} \\ c_k &= c_0 + N_k \\ a_k &= a_0 + N_k \\ b_k &= b_0 + S + \tfrac{c_0 N_k}{c_0 + N_k} (\bar{x}_k - m_0)(\bar{x}_k - m_0)^T \end{aligned} \tag{2.23}$$

where $\bar{x}_k$, N_k and S are computed in the same way as in Section 2.3.2. Note that while (m_0, c_0, b_0, a_0) denote the prior terms of the NIW distribution and should be specified a priori, (m_k, c_k, b_k, a_k) are the corresponding posterior terms of the NIW posterior that are estimated using the data and the information from the prior. The optimization of the variational posterior distribution then involves first updating the indicators from the

expectation:

$$\mathbb{E}\left[z_{ik}\right] = \exp\left(\mathbb{E}\left[\ln \pi_k\right] + \frac{\mathbb{E}\left[\ln\left|\Sigma_k^{-1}\right|\right]}{2} - \frac{\mathbb{E}_{\mu_k,\Sigma_k^{-1}}\left[\left(x_i-\mu_k\right)^T \Sigma_k^{-1}\left(x_i-\mu_k\right)\right]}{2}\right) \tag{2.24}$$

where we use

$$\mathbb{E}_{\mu_k,\Sigma_k}\left[\left(x_i-\mu_k\right)^T \Sigma_k^{-1}\left(x_i-\mu_k\right)\right] = Dc_k^{-1} + a_k\left(x_i-m_k\right)^T b_k\left(x_i-m_k\right) \tag{2.25}$$

$$\mathbb{E}\left[\ln\left|\Sigma_k^{-1}\right|\right] = \sum_{d=1}^{D}\psi\left(\frac{a_k+1-d}{2}\right) + D\ln\left(2\right) + \ln\left|b_k\right| \tag{2.26}$$

$$\mathbb{E}\left[\ln \pi_k\right] = \psi\left(\alpha_k\right) - \psi\left(\hat{\alpha}\right) \tag{2.27}$$

where $\hat{\alpha} = \sum_k \alpha_k$ and $\psi\left(\cdot\right)$ is the digamma function. Then we hold the indicators fixed while updating the variational posterior of the component parameters π, μ and Σ using the the expressions for $q\left(\pi\right)$ and $q\left(\mu_k,\Sigma_k\right)$ from above. We iterate between the two (in E-M fashion) until convergence at a fixed solution is reached.

2.3.4 Iterated conditional modes

Despite the asymptotic guarantees of the Gibbs sampler, for mixture models it can take rather long to converge to a good estimate of the posterior. On the other hand, the K-means algorithm makes some rather restrictive assumptions in addition to those used to define the GMM, which can lead to surprising behaviour. However, a clustering algorithm almost as simple as K-means can be derived from the GMM, that overcomes a lot of the pitfalls of K-means while also remaining simple and scaling well. The basic idea is to use *conditional modal point estimates* rather than samples from the conditional probabilities as in Gibbs sampling. This is a simple *maximum-a-posteriori* (MAP) method that can be used to derive algorithms which converge to at least locally optimal model fits to the data. Unlike MCMC methods and E-M, such methods do not provide any information about the uncertainty of the model fit, or the distribution of the latent variables. The approach is known as *iterated conditional modes* (ICM) (Besag, 1986), later also called *the maximization-maximization* (M-M) algorithm by (Welling & Kurihara, 2006). Consider the GMM from (2.9) as a starting point, iterative MAP computes for each $i \in \{1,\ldots,N\}$ the negative logarithm of each $p\left(z_i = k\,|\mu_k,\Sigma_k,\pi_k,x\right)$ from (2.13):

$$d_{i,k} = \frac{1}{2\Sigma_k}\left\|x_i-\mu_k\right\|_2^2 + \frac{D}{2}\ln\Sigma_k - \ln\pi_k \tag{2.28}$$

for $k = 1,\ldots,K$ where we can ignore normalization terms which are independent of k. We update z_i such that:

$$z_i = \underset{k\in\{1,\ldots,K\}}{\arg\min}\; d_{i,k} \tag{2.29}$$

The algorithm proceeds with updating the model parameters with the values that maximize their corresponding posterior distributions. For each component k we update the cluster means and covariances using the mode of the NIW posterior:

$$\begin{aligned}\mu_k &= m_k \\ \Sigma_k &= b_k\left(a_k - D - 1\right)\end{aligned} \tag{2.30}$$

with m_k, b_k and a_k computed as in (2.15). Finally, we update the component weights using the closed form mode of the Dirichlet posterior from Equation (2.16):

$$\pi_k = \frac{N_k + {}^{\alpha}/_{K} - 1}{N + \alpha - K} \tag{2.31}$$

for $k \in \{1, \ldots, K\}$. The procedure iterates between updating the assignments z, the cluster parameters (μ, Σ) and the mixing weights π until convergence. Similar to K-means, this iterative MAP inference is guaranteed to converge to a local clustering solution, but unlike K-means it actually optimizes the complete negative log-likelihood of the Bayesian GMM, (2.17). That is, at each iteration MAP does not increase $-\ln\left(p\left(x, z|\mu, \Sigma, \pi\right)\right)$ and eventually converges to a fixed point clustering solution.

In Algorithm **2.2** we summarize the ICM method applied to the simple GMM with spherical component covariances. This setup allows us to easily see that spherical MAP introduces only slight modifications to the K-means algorithm, but we can resolve some of its drawbacks: for example the data is no longer clustered purely based on its geometric closeness, see Issue 3 from Section 2.1. Distances are now also balanced with the relative cluster densities reflected through the mixture weight term π_k and the σ term allows for explicit control over component spread.

The fact that ICM optimizes the original likelihood of the probabilistic model enables us to use it (or equivalently the negative log likelihood) for model selection and predictions. This means that we can compare the quality of different clustering solutions, produced using ICM applied to the Bayesian GMM, by comparing the corresponding NLL values (for the Bayesian GMM this involves computing (2.17)). We discuss strategies for making predictions using ICM in later sections. The execution time of K-means and MAP is comparable and can be a few orders of magnitude less than with corresponding MCMC methods such as Gibbs.

2.4 Marginalization

The process of *collapsing* a probabilistic model involves integrating out random variables from the probabilistic model to improve computational efficiency of an associated inference algorithm. Collapsing in Bayesian inference is only a specific application of a more general technique known as *Rao-Blackwellization* (Kolmogorov, 1950). In this technique we use sufficient statistics of an unknown quantity to improve its statistical estimators. For example, let $p\left(x, \theta\right)$ denotes some target distribution on two random variables $x \in \Omega$ and $\theta \in \Theta$ and $\{(x_i, \theta_i)\}_{i=1}^{N}$ are N independent draws from this joint distribution. Probably the simplest way to estimate some statistic $f\left(x, \theta\right)$ is using:

$$\begin{aligned} \mathbb{E}_p\left[f\left(x, \theta\right)\right] = & \int_{\Theta}\int_{\Omega} f\left(x, \theta\right) p\left(x, \theta\right) dx\, d\theta \\ \approx & \frac{1}{N}\sum_{i=1}^{N} f\left(x_i, \theta_i\right) = \mathbb{E}_{\tilde{p}}\left[f\left(x, \theta\right)\right] \end{aligned} \tag{2.32}$$

where $\mathbb{E}_{\tilde{p}}\left[f\left(x, \theta\right)\right]$ is an estimator of $\mathbb{E}_p\left[f\left(x, \theta\right)\right]$ (hence the notation $\tilde{p}\left(\cdot\right)$ which is an estimator of the target distribution). Now, consider that the conditional distribution $p\left(x\,|\theta\right)$ is available in some tractable analytic

closed form. Then, we can consider the following alternative estimator:

$$\begin{aligned}\mathbb{E}_p\left[f\left(x,\theta\right)\right] = & \int_\Theta \int_\Omega f\left(x,\theta\right) p\left(x\left|\theta\right.\right) p\left(\theta\right) dx\, d\theta \\ = & \int_\Theta \left[\int_\Omega f\left(x,\theta\right) p\left(x\left|\theta\right.\right) dx\right] p\left(\theta\right) d\theta \\ \approx & \frac{1}{N}\sum_{i=1}^{N} \int_\Omega f\left(x,\theta_i\right) p\left(x\left|\theta_i\right.\right) dx = \mathbb{E}_{\tilde{p}}\left[\mathbb{E}_p\left[f\left(x,\theta\right)\left|\theta\right.\right]\right]\end{aligned} \tag{2.33}$$

Both estimator $\mathbb{E}_{\tilde{p}}\left[f\left(x,\theta\right)\right]$ and estimator $\mathbb{E}_{\tilde{p}}\left[\mathbb{E}_p\left[f\left(x,\theta\right)\left|\theta\right.\right]\right]$ are unbiased and asymptotically converge to $\mathbb{E}_p\left[f\left(x,\theta\right)\right]$ almost surely as $N \to \infty$. However the conditional estimator $\mathbb{E}_{\tilde{p}}\left[\mathbb{E}_p\left[f\left(x,\theta\right)\left|\theta\right.\right]\right]$ can be thought of as more reliable, because the inherent sample space of Ω is smaller than the original space $\Omega \times \Theta$. Proof that starting from an unbiased estimator, the estimator's variance can be reduced by conditioning simpler estimators with respect to some appropriate statistic can be found in (Sudderth, 2006).

Theorem: Rao-Blackwell Let x and θ be dependent random variables and $f\left(x,\theta\right)$ a scalar statistic. Consider the marginalized statistic $\mathbb{E}_x\left[f\left(x,\theta\right)\left|\theta\right.\right]$, which is a function solely of θ. The unconditional variance $\mathrm{Var}_{x,\theta}\left[f\left(x,\theta\right)\right]$ is then related to the variance of the marginalized statistic as follows:

$$\begin{aligned}\mathrm{Var}_{x,\theta}\left[f\left(x,\theta\right)\right] = & \mathrm{Var}_\theta\left[\mathbb{E}_x\left[f\left(x,\theta\right)\left|\theta\right.\right]\right] + \mathbb{E}_\theta\left[\mathrm{Var}_x\left[f\left(x,\theta\right)\left|\theta\right.\right]\right] \\ \geq & \mathrm{Var}_\theta\left[\mathbb{E}_x\left[f\left(x,\theta\right)\left|\theta\right.\right]\right]\end{aligned} \tag{2.34}$$

From (2.34) it follows that analytic marginalization of some variables from a joint distribution will always reduce the variance of marginal estimates. Furthermore, it will also follow that integrating over some θ is most useful when the average variance of x conditional on θ is large. This variance reduction, guaranteed by the theorem, also generalizes to estimates based on the correlated samples produced by a Gibbs sampler (Liu *et al.*, 1994).

2.5 The Dirichlet process

In Section 2.3 we introduced Bayesian finite mixture models (and important ideas related to inference in them) as a flexible probabilistic way to model clustering of observed data. However, finite mixture models require a fixed number of mixture components K to be chosen. Model selection using the complete data likelihood of the model is a rigorous approach to tuning the hyperparameters of a Bayesian mixture, but formally K is not a hyperparameter. In the Bayesian context, this is because K has not been treated as random. Placing a prior distribution over K or comparing different solutions for the model using different values of K can be inefficient and often lead to incorrect results in practice. For many applications, an adequate model should be able to learn the number of components based on the available data, adapting as more data becomes available. This will be the underlying motivation for constructing the Bayesian *infinite* mixture model, i.e. the Dirichlet process mixture model. The infinite mixture model replaces the assumption that there are a fixed, finite number of mixture components with a one where there is an infinite number of possible components and for any finite dataset only some unknown finite number of them are represented. To understand better the construction and the properties of such models we start by introducing here the DP and some of its most useful properties and constructions.

2.5.1 Definition

Let (Θ, B) be a measurable space[3], with G_0 a probability measure[4] on the space. Let α be a positive real number. A *Dirichlet process*, $\mathrm{DP}(\alpha, G_0)$, is defined as the distribution of a *random probability measure* G over (Θ, B) such that, for any finite measurable partition $(A_1, ..., A_K)$ of Θ, the random vector $(G(A_1), ..., G(A_K))$ is distributed as a finite dimensional Dirichlet distribution with parameters $(\alpha G_0(A_1), ..., \alpha G_0(A_K))$, i.e.:

$$G(A_1), ..., G(A_K) \sim \mathrm{Dirichlet}(\alpha G_0(A_1), ..., \alpha G_0(A_K)) \tag{2.35}$$

for any $A_1 \cup \cdots \cup A_K \equiv \Theta$(i.e. partition of the set into exhaustive and mutually exclusive subsets of the elements of that set). That is, we say a probability mass function G is a draw from a DP if all of its possible finite marginal distributions are Dirichlet distributed.

A DP is parametrized by a *concentration parameter* $\alpha > 0$ and some *base distribution* G_0 where the base distribution is the expectation of the DP. For $G \sim \mathrm{DP}(\alpha, G_0)$, the base distribution G_0 specifies the expectation of the probability measure G, $\mathbb{E}[G] = G_0$. This is straightforward to see if we use (2.35) and we substitute concentration parameter $\alpha = 1$, from the properties of the Dirichlet distribution it follows that $\mathbb{E}[G(A_k)] = G_0(A_k)$ for $k = 1, \ldots, K$. Therefore, G_0 effectively specifies where the probability mass of draw G is distributed on average. Note that G_0 can be both continuous or discrete, unlike G which is forced to be discrete. The parameter α reflects our belief in the base distribution G_0, or the concentration of G around G_0. For some fixed G_0, as $\alpha \to 0$ the draw $(G(A_1), G(A_2), \ldots, G(A_K))$ will get *sparser* as (from (2.35)) it would be drawn from a Dirichlet distribution with diminishing α and the mass will be concentrated on a single subset A_i, with the rest of Θ having nearly 0 support. As $\alpha \to \infty$ the distribution of $(G(A_1), \ldots, G(A_K))$ will get closer to $(G_0(A_1), G_0(A_2), \ldots, G_0(A_K))$.

As discussed above, a G drawn from a $\mathrm{DP}(\alpha, G_0)$ is a probability distribution and we can draw samples ϕ_i from it. We can then discuss the properties and form of such a DP-distributed measure G in terms of some draws from it $\phi_1, \ldots, \phi_N$. Due to the discreteness G, each draw ϕ will take exactly one of some K values, with the probability of the draws in full generality being categorically distributed. For any partition $(A_1, \ldots, A_K)$, we use N_k to denote the number of ϕ's in A_k, $N_k = \sum_{i:\phi_i \in A_k} 1$. Then using the conjugacy between the Dirichlet and the categorical distributions, we can write the posterior of any finite marginal of G as:

$$(G(A_1), \ldots, G(A_K)) \sim \mathrm{Dirichlet}(\alpha G_0(A_1) + N_1, \ldots, \alpha G_0(A_K) + N_K) \tag{2.36}$$

and this will hold for any partition $(A_1, \ldots, A_K)$ of Θ. Using the defining property of a DP (2.35), we can say that the posterior of the measure G, $p(G|\phi_1, \ldots, \phi_N)$ is also a DP and we will now derive the posterior parameters of that DP.

The samples ϕ from a DP-distributed G have a strong *clustering* property that they share with all discrete distributions: draws $\phi_1, \ldots, \phi_N$ have repeated values and we will denote the unique values that they take with $\theta_1, \ldots, \theta_K$ where usually $N \gg K$. It will then be beneficial to choose a partition which places draws sharing the same value θ_k in the same subset and further split samples with different values into different subsets. Formally, choose a partition $(A_1, \ldots, A_K, A_{K+1}, \ldots, A_{K+\nu})$ for some integer $\nu > K$ such that $A_k = \{\phi_i : \phi_i = \theta_k, i \in \{1, \ldots, N\}\}$ for $k = 1, \ldots, K$ and $A_{K+1} \cup \cdots \cup A_{K+\nu} = \Theta \backslash A_1 \backslash \ldots \backslash A_K$. From the

[3] A measurable space is some pair (Θ, B) consisting of a set B and a σ-algebra Θ of subsets of B. The σ-algebra Θ is a collection of subsets of B that includes the empty subset, is closed under complement, and is closed under countable unions and countable intersections.

[4] A probability measure is a measure which assigns the value 1 to the entire probability space.

definition of a Dirichlet process and the conjugacy of the Dirichlet and the categorical distributions we have:

$$\begin{aligned}(G(A_1),\ldots,G(A_{K+1})) \sim & \text{Dirichlet}(\alpha G_0(A_1)+N_1,\ldots,\alpha G_0(A_K)+N_K,\alpha G_0(A_{K+1}),\ldots,\alpha G_0(A_{K+\nu})) \\ \sim & \quad \text{Dirichlet}(N_1,\ldots,N_K,\alpha G_0(A_{K+1}),\ldots,\alpha G_0(A_{K+\nu}))\end{aligned} \tag{2.37}$$

Since this holds for any partition of the space Θ, this is by definition a DP but with updated concentration parameter $\alpha+N$ and updated base distribution:

$$\frac{\alpha G_0+\sum_{i=1}^N \delta_{\phi_i}}{\alpha+N}. \tag{2.38}$$

Equation (2.36) allows us to compute the predictive likelihood $p(\phi_{N+1}|\phi_1,\ldots,\phi_N)=\int p(\phi_{N+1}|G)\,p(G|\phi_1,\ldots,\phi_N)\,dG$ as follows. If we choose any set $A_k \subsetneqq \Theta$ with $k\in\{1,\ldots,K,\ldots\}$ and let us compute $p(\phi_{N+1}\in A_k|\phi_1,\ldots,\phi_N)$. We integrate out the DP G and find:

$$\begin{aligned}p(\phi_{N+1}\in A_k|\phi_1,\ldots,\phi_N) &= \int p(\phi_{N+1}\in A_k|G)\,p(G|\phi_1,\ldots,\phi_N)\,dG \\ &= \int G(A_k)\,p(G|\phi_1,\ldots,\phi_N)\,dG \\ &= \mathbb{E}[G(A_k)|\phi_1,\ldots,\phi_N] \\ &= \frac{1}{\alpha+N}\left(\alpha G_0(A_k)+\sum_{i=1}^N \delta_{\phi_i}(A_k)\right) \\ &= \frac{\alpha}{\alpha+N}G_0(A_k)+\sum_{j=1}^K \frac{N_j}{\alpha+N}\delta_{\theta_j}(A_k)\end{aligned} \tag{2.39}$$

If we let $A_k=\Theta\backslash\{\phi_1,\ldots,\phi_N\}$, then we find ϕ_{N+1} is drawn from the base measure G_0 with probability $\frac{\alpha}{\alpha+N}$. If we let $A_k=\{\theta_k\}$ then $\phi_{N+1}=\theta_k$ with probability $\frac{N_k}{\alpha+N}$. This means that with probability $\frac{N_k}{\alpha+N}$ the new draw will be equal to an existing θ_k. This argument makes it clear that draws from G cluster together around the same θ_k; hence we will often refer to the θ_k as *atoms* of the distribution G. We will see that those predictive probabilities are exactly the same as those defined by the *Chinese restaurant process* later on in Section (2.5.2). In fact an important property of this stochastic process is that it defines the distribution of partitions of samples from a DP-drawn random measure G.

2.5.2 Constructions

Stick-breaking construction

Probability measures G drawn from a DP have the interesting property that they are *discrete with probability one*, or any draws from a DP, regardless of the sample space of the base distribution, are discrete by definition. Therefore, one way to look at discrete probability density G is as a mixture of point masses i.e. $G=\sum_{k=1}^\infty \pi_k\delta_{\theta_k}$ where δ_{θ_k} is the (multidimensional) Dirac delta function centered at θ_k. If G is a draw from a DP, $G\sim \text{DP}(\alpha,G_0)$, then the θ_k should be independent and identically distributed draws from the base distribution G_0, $\theta_k\sim G_0$ for $k=1,2,\ldots$. We can think of the point masses as the discrete domain of values of G and the mixing coefficients π_k as the support of each discrete atom, where the the mixing coefficients sum to 1, $\sum_{k=1}^\infty \pi_k=1$. The mixing coefficients should be constructed using a *stick-breaking* construction as Sethuraman (1994) showed. Suppose there is a stick with length 1. At each step k, for $k=1,2,3,\ldots$, we take $\beta_k\sim\text{Beta}(1,\alpha)$ part from what is left of the stick, then π_k can be calculated as the length we actually

at each step:

$$\pi_1 = \beta_1, \pi_2 = (1-\beta_1)\,\beta_2, \ldots, \pi_k = \beta_k \prod_{j=1}^{k-1} (1-\beta_j), \ldots, \tag{2.40}$$

For the weights $\pi_1, \ldots, \pi_\infty$ generated using the stick-breaking process with parameter α we will write $\pi \sim \text{Stick}(\alpha)$. The stick-breaking constriction is one view of generating draws from a DP using the discreteness of its mass function (Gershman & Blei, 2012). Note that the density G drawn from a DP is represented as an infinite sum and so does not have a closed form probability density function, this is one of the problems we have to overcome when dealing with the DP (Gershman & Blei, 2012). However, the stick-breaking construction provides us with the steps to accurately generate an unlimited number of mixing coefficients π_k in this sum. The atoms θ_k can be obtained by sampling from the known base distribution G_0. The importance of the stick-breaking approach for this thesis will come from the fact that using this construction, we will be able to define tractable inference for the Dirichlet process mixture model (and other BNP models such as HDP and HDP-HMM from later chapters) that does not require integration over the random DP distributed measure G. Neal (2003) introduced a *slice sampling* method for inference in DP mixtures, which asymptotically recovers the exact DP mixture model posterior over the full parameters space, avoiding any collapsing. Inference methods for DP mixtures, which are based on the Chinese restaurant process construction will always require integrating out G. Following up on our discussion on marginalization in probabilistic Bayesian models, the stick-breaking construction will allow us to derive memory efficient algorithms and will also simplify inference in non-conjugate DP-based probabilistic models.

Polya urn construction

The *Polya urn scheme* does not refer to G directly; rather, it refers to draws from G. Thus let $\phi_1, ..., \phi_N$ be a sequence of independent and identically distributed random variables drawn from G. That is, the variables $\phi_1, ..., \phi_N$ are conditionally independent given G, and hence are *exchangeable* (their joint distribution is independent of any ordering). Let us consider the successive conditional distributions of ϕ_i given $\phi_1, ..., \phi_{N-1}$, where G has been integrated out. Blackwell (1947) showed that these conditional distributions have the following form:

$$p\left(\phi_{N+1} \,|\phi_1, ..., \phi_N, \alpha, G_0\right) \sim \sum_{i=1}^{N} \frac{1}{N+\alpha}\delta_{\phi_i} + \frac{\alpha}{N+\alpha} G_0 \tag{2.41}$$

We can interpret this conditional distribution in terms of a simple *urn model* in which a ball of a distinct colour is associated with each distinct θ_i. The balls are drawn with equal probability; when a ball is drawn, it is placed back in the urn together with another ball of the same color, i.e. the same value θ_i. In addition, with probability proportional to α, a new colour ball is created by drawing from G_0, and this ball is added to the urn.

In the Polya urn scheme, we overcome the problem of constructing an infinite mixture of point masses G, by integrating over G and dealing instead with the successive conditional probabilities of the draws of G, derived in (Blackwell, 1947). Where the conditional probability of θ_N given G is an infinite mixture, the marginal likelihood of θ_N is available in closed form (2.41). This exploitation of marginalization is widely used when dealing with Bayesian nonparametric models since the inherently infinite parameter space can be integrated out and the marginal structure can be used for more efficient inference.

Chinese restaurant process (CRP)

A closely related stochastic process to the Polya urn scheme is the *Chinese restaurant process* (CRP). In fact, the CRP defines the distribution over partitions that is inherent in the partitioning of the elements $\theta_1, \ldots, \theta_N$ of the Polya urn scheme. Furthermore, this is the partition implied by the data after integrating out the mixing distribution G from a DPMM - the CRP is a marginal process of the DP.

Recall from above that by *partition* we mean the usual concept of a set of subsets of some items, where each item is allowed to belong to exactly one subset. The set of subsets is exhaustive and mutually exclusive. The subsets are also referred to as clusters. Let us denote a partition of N items as τ_N. We are going to denote the number of subsets (clusters) in τ_N as K and we will index the different subsets in τ_N with k where $k \in \{1, ..., K\}$. Further $c_1, ..., c_K$ will refer to the different subsets in partition τ_N. For consistency with the notation in the rest of the report, we will denote the cardinality of cluster k as $N_k = |c_k|$. Let us consider we have N items $x_1, ..., x_N$, then τ_N would be some exhaustive and mutually exclusive set of subsets of $x_1, ..., x_N$. For example, if $N = 8$, a possible partition would be $\tau_8 = \{\{x_1, x_2\}, \{x_3, x_5, x_7\}, \{x_4, x_6\}, \{x_8\}\}$. The total number of subsets in τ_8 is $K = 4$, and c_3 refers to the subset of elements $\{x_4, x_6\}$ and $N_3 = 2$. Note that the subsets can instead be constructed using sets of indices $i = 1, 2, \ldots$, so that, e.g. $c_3 = \{4, 6\}$, which is the notation we adopt from now on. Furthermore, introducing subset assignment variables $z_1, \ldots, z_8$, we can write $c_3 = \{x_i : \forall x_i \,|z_i = 3, i \in \{1, \ldots, 8\}\}$.

The CRP is a probability distribution on partitions parametrized by concentration parameter α and number of elements in the partition N. To explain further the behaviour and properties of this distribution and how is it determined by those parameters, let us show how to construct partitions that are CRP-distributed. We will start from a one point partition (one subset with one element in it) and will add one point at a time in the partition, sampling subset assignments for new points from a conditional rule; after increasing the points in the partition up to N, we will obtain a partition that is a draw from the CRP for N points. This defining conditional rule depends on previous subset assignments and the concentration parameter α and is often described using the metaphor of a restaurant, with data points corresponding to customers and subsets (clusters) corresponding to tables. Customers arrive at the restaurant one at a time. The first customer is seated alone. Each subsequent customer is either seated at one of the already occupied tables with probability proportional to the number of customers already seated there, or, with probability proportional to the parameter α, the customer sits at a new table. An important point to make is that the tables are to be viewed as unordered and so permutations of tables in the restaurant does not change the partition. To avoid introducing labels that suggest an ordering, we refer to a table by the subset of customers sitting at the table. We keep the same notation from above and use k to denote a cluster index and c_k to denote the cluster itself, similarly N_k would denote the number of customers sitting at table c_k. With this metaphor and notation, we write the probabilistic rule characterizing the CRP as follows:

$$P\left(\text{customer } N+1 \text{ joins table } c_k\right) = \begin{cases} \frac{N_k}{\alpha+N} & \text{if } c_k \in \tau_N \\ \frac{\alpha}{\alpha+N} & \text{otherwise} \end{cases} \tag{2.42}$$

After N customers have arrived, their seating pattern defines a set of clusters and thus a partition. This partition is random, and thus the CRP is a distribution on partitions. From now on we will denote this as:

$$\tau_N \sim \text{CRP}\left(\alpha, N\right)$$

Further, we can compute $P(\tau_N)$ given that τ_N is a draw from a CRP by:

$$P(\tau_N) = \frac{\alpha^K}{\alpha^{(N)}} \prod_{c_k \in \tau_N} (N_k - 1)! \tag{2.43}$$

where $\alpha^{(N)} = \alpha(\alpha+1)...(\alpha+N-1)$ and we will use this notation for the whole section. This probability is obtained as a product of the probabilities in (2.42). There are exactly K number of tables in τ_N and so customers have sat on a new table K times, explaining the term α^K in the expression. The probability of a customer sitting on an existing table c_k is used $N_k - 1$ times where each time the numerator of the corresponding probability increases, from 1 to $N_k - 1$. This is how the term $\prod_{c_k \in \tau_N} (N_k - 1)!$ in the probability of the partition arises. The $\alpha^{(N)}$ is the product of the denominators when multiplying the probabilities from (2.42), as $N = 1$ at the start and increases to $N - 1$ for the last seated customer. Therefore, we can see that the parameter α controls the rate of increase of the number of subsets in τ_N as N increases. It is usually referred to as the concentration parameter because it determines the concentration of customers around the tables. If we hold the number of customers fixed, as α increases the number of subsets K (tables) in τ_N increases.

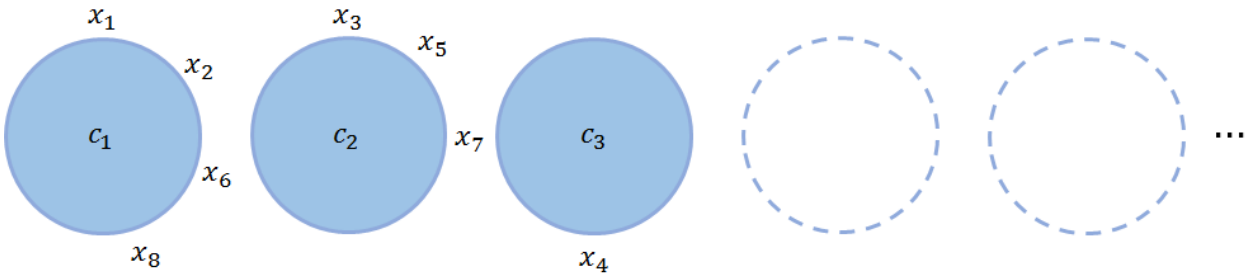


Figure 2.2: The number of available tables R, which includes both the occupied and the empty (dashed) ones is infinite, while the number of occupied tables, K, is finite and unknown for any finite number of customers $x_1, \ldots, x_N$.

Despite the CRP being specified by a partition of items, an important property of this distribution on partitions is that it is invariant to the ordering, i.e. only the *size* of the clusters matters in determining the probability of the partition, not the identities of the specific items forming the clusters. This is an example of probabilistic exchangeability and is essential to using the CRP as a constructive mechanism for mixture models later on. Exchangeability is a natural property for clustering data and many algorithms, including K-means defined earlier, are invariant to the ordering of the data points.

2.5.3 Pitman-Yor generalization

The *Pitman-Yor process* (PYP) (Pitman & Yor, 1997) is a generalization of the DP, which has an additional *discount hyper-parameter* that gives the PYP more control than the DP over the rate at which new clusters are created. The expected number of clusters to be created by a DP is approximately $\alpha \log N$ which grows slowly with N (Sammut & Webb, 2011, page 285). So, with more data arriving, the model might create fewer clusters than we might wish. For many applications we might wish the number of clusters to be able to grow more quickly. Then the PYP turns out to be useful. It can be shown that the expected number of clusters in the PYP with *discount parameter* κ is proportional to N^κ (Pitman & Yor, 1997). In addition, it allows for more control than the DP over the tails of the partition distribution it implies, i.e. more control over the support of the smaller clusters. While the distribution that the DP defines over the size of the clusters to

be created has exponential tails, the PYP allows us to model cluster sizes with power-law tail distribution. A PYP corresponds to a distribution over a discrete probability distribution $G \sim \text{PY}(\alpha, \kappa, H)$. The hyper parameters $\alpha > -\kappa$ and $0 \leq \kappa < 1$ control the distribution over the mixing weights π (using same notation as above). The base distribution G_0 is a prior over the parameter space Θ. A draw from $\text{PY}(\alpha, \kappa, G_0)$ has the following representation: $G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$. The weights can be obtained with a (modified) stick-breaking process:

$$\begin{aligned} \beta_k &\sim \text{Beta}(1-\kappa, \alpha + k\kappa) \\ \pi_k &= \beta_k \prod_{j=1}^{k-1} (1-\beta_j) \end{aligned}$$

The parameters are distributed as $\theta_k \sim G_0$. Then if $\theta_1, ..., \theta_N \sim G$ and $G \sim \text{PY}(\alpha, \lambda, G_0)$ we will be interested in the following conditional (marginal) probabilities:

$$\theta_N | \theta_1, ..., \theta_{N-1} \sim \sum_{k=1}^{K} \frac{N_k - \kappa}{\alpha + N - 1} \delta_{\theta_k} + \frac{\alpha + K\kappa}{\alpha + N - 1} G_0 \tag{2.44}$$

where N_k counts how many parameter sets θ have been sampled from component k. All of the clustering algorithms derived in this thesis from the DP mixture model, can be easily extended to Pitman-Yor mixture-based algorithms that will simply have an additional parameter κ, giving additional control over how quickly the number of clusters increase with the data.

2.5.4 Dirichlet process mixture models

Infinite mixture models

An intuitive explanation that can help us understand the assumptions inherent in the DP mixture model is to view it as the limiting process obtained by assuming a Bayesian mixture model with K components and allowing $K \to \infty$, also known as the *infinite mixture model* (Rasmussen, 1999). Recall that in finite mixture models, the conjugate prior over the mixing coefficients was $\pi_1, \ldots, \pi_K \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$ where here for ease of notation, we consider a simple uniform Dirichlet prior with parameters $\alpha_1 = \cdots = \alpha_K = \alpha/K$. Therefore, a priori we place some non-zero support on some fixed K number of components. From conjugacy it follows that the corresponding posterior of π is also a Dirichlet distribution, with $\boldsymbol{\pi} \sim \text{Dirichlet}(N_1 + \alpha/K, ..., N_K + \alpha/K)$. This means that the π_k reflect a trade-off between the number of observations already assigned to a component and the prior quantity α/K, where if no observations belong to a group there is still some non-zero support α/K on that component.

However, if we assume the number of components $K \to \infty$, the prior support on any of those infinitely many components goes to zero at the limit, $\alpha/K \to 0$. In other words, there exist support on infinitely many components, but the probability value of that support is 0 at the limit. Let us denote with K^+ the number of components which are actually represented after observing the data. In the Bayesian finite mixture model (Section 2.3) by design we always have $K^+ = K$ because each component has non-zero prior support α/K. As $K \to \infty$ this is no longer guaranteed.

Unlike K which is the number of components in the 'latent' structure of the model, which could also be infinite, K^+ is forced to be finite as we always have a finite amount of data. If we assume that the size of the data is infinite then asymptotically $K^+ \to K$, but for practical purposes when the data is finite, so will be K^+.

For each of the components represented in the data, the counts N_k of observations assigned to it is positive. Also the support for π_k becomes proportional to N_k for represented components as $N_k + \alpha/K \to N_k$ for $K \to \infty$. Consider now the remaining $K - K^+$ non-represented components. There are infinitely many, as $\lim_{K\to\infty} K - K^+ \to \infty$ each one with 0 probability of having an observation assigned to them. However, let us now look at the probability of assigning an observation to *any* of these non-represented components. Due to the *aggregation property* of the Dirichlet distribution, the posterior Dirichlet parameter value associated with all non-represented components lumped together, will be:

$$\sum_{k=1}^{K-K^+} \frac{\alpha}{K} = \frac{K - K^+}{K}\alpha \tag{2.45}$$

We can see that $\lim_{K\to\infty} \frac{K-K^+}{K} = 1$ for K^+ is finite. Then the probability of assigning an observation to any of the non-represented components is proportional to α, so that only by combining all non-represented component weights we do get non-zero combined weight $\pi_{K^++1} \propto \alpha$ which holds for any finite K^+. In the rest of the thesis we use notation K for both number of population clusters for finite models and the unknown number of represented components. Furthermore, by clarifying the difference in the meaning of the concentration parameter α for the Dirichlet distribution and the Dirichlet process, we can think of the concentration parameter as a prior *item count* and denote it with N_0, having $\alpha \equiv N_0$ in all of the following figures and expressions throughout the thesis.

CRP mixture model

The CRP provides us with a model for probabilities over partitions, but leaves us short of a model for generating data points from a mixture. To take the next step, we will derive a model that generates data points that cluster in partitions which are distributed according to a CRP. To each table c_k in the partition τ_N (more formally $\tau_N = c_1 \cup \cdots \cup c_K$) we assign a parameter vector θ_k and we make the assumption that the data points from table c_k are generated independently from a common probability distribution with parameters θ_k. Then, as is typical for mixture models, let $f(x_i|\theta_k)$ denote the probability density for generating data point x_i from the distribution parametrized by θ_k. For instance, if we use a Gaussian the probability density for point x_i from cluster c_k would be $f(x_i|\theta_k) = \mathcal{N}(x_i|\mu_k, \Sigma_k)$ with $\theta_k = \{\mu_k, \Sigma_k\}$. Now that we have a generating distribution for single observations and with known cluster assignments, we can obtain the overall likelihood of the data. The overall conditional probability of the data is the product over clusters and over data points within clusters:

$$p(x|\theta, \tau_N) = \prod_{c_k \in \tau_N} \prod_{i:z_i=k} f(x_i|\theta_k) \tag{2.46}$$

To provide a complete Bayesian probabilistic model, we need to specify a prior over the parameters θ, and we denote this as G_0. Then the model generating the data points is:

$$\begin{aligned} \tau_N &\sim \text{CRP}(N_0, N) \\ \theta_k &\sim G_0 \text{ for } c_k \in \tau_N \\ x_i &\sim F(\theta_k) \text{ for } c_k \in \tau_N,\ i : z_i = k \end{aligned} \tag{2.47}$$

where $F(\theta_k)$ denotes the distribution defined with probability density function $f(x|\theta_k)$ and we have used N_0 instead of α for notation of the concentration parameter (prior item count). These linked conditional

probabilities yield a joint probability distribution on the set of variables (x, θ, τ_N) and then Bayesian inference can be invoked to obtain various posterior probabilities of interest, in particular the probability of the partitions given the data, which serves as a method of clustering the data. We can also use this construction to sample data that induces a CRP partition. Consider that $F(\theta_k)$ is Gaussian and we choose the conjugate Normal-Gamma (NG) prior over all θ_k (the variables in each dimension of θ_k, $\theta_{k,1}, \ldots, \theta_{k,D}$ are independent). Then if we sample component parameters from the NG prior with $m_0 = [1, 1]$, $c_0 = 1$, $b_0 = [0.5, 0.5]$, $a_0 = 1$ and cluster indicators are sampled from a CRP ($N_0 = 3$, $N = 600$), the data will take the form in Figure 2.3. As expected, when using a CRP prior, the sizes of the different clusters vary significantly with many small clusters containing only a few observations in them.

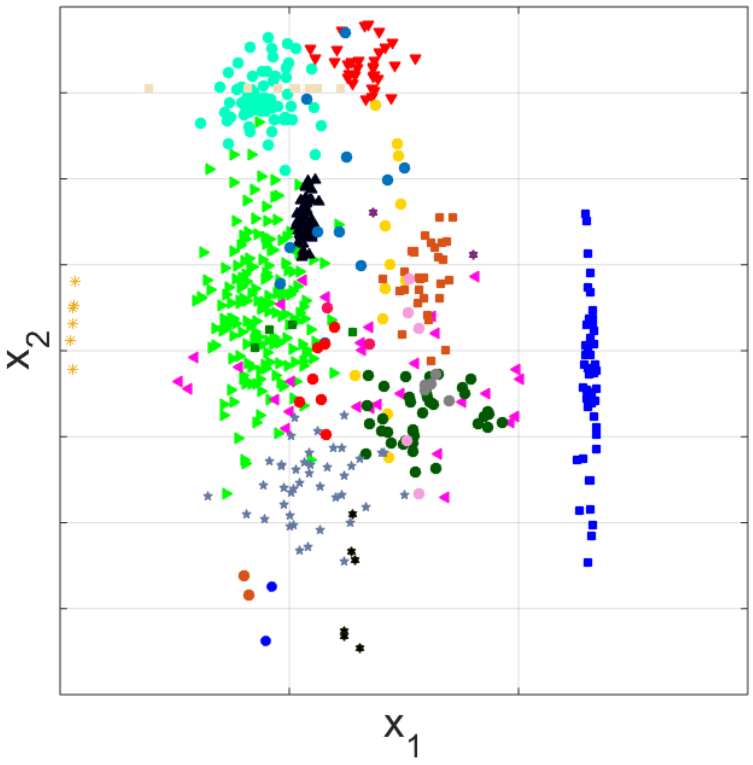


Figure 2.3: Sample from $D = 2$ CRP probabilistic model containing $K = 22$ clusters ranging in size from the biggest cluster $N_k = 180$ to two clusters with $N_k = 2$.

2.6 Overview of the relations between inference algorithms

In this chapter we introduced some of the fundamental concepts behind mixture models and how they can be motivated as a probabilistic way of describing clustered data. These concepts in later chapters serve as building blocks for deriving a set of novel inference algorithms which can overcome a lot of the practical challenges posed by current algorithms for inference in discrete latent variable models.

In order to better navigate the reader to how those novel methods are derived in the following chapters, in Figure 2.4 we organize a chart which explicitly displays the relations between most of the techniques presented in the thesis. We have used four main criteria to classify those relations:

- Parametric/Nonparametric - inference algorithms based on the parametric and nonparametric version of the same probabilistic model.

- Collapsed/Non-collapsed - inference algorithms based on the complete and collapsed construction[5] of the same probabilistic model.
- ICM - iterated conditional models inference algorithms for a given probabilistic model.
- SVA - 'K-means like' methods which can be seen as *small variance asymptotic* inference algorithms for a given probabilistic model. Small variance asymptotic reasoning is described in more detail later in Section 3.2.

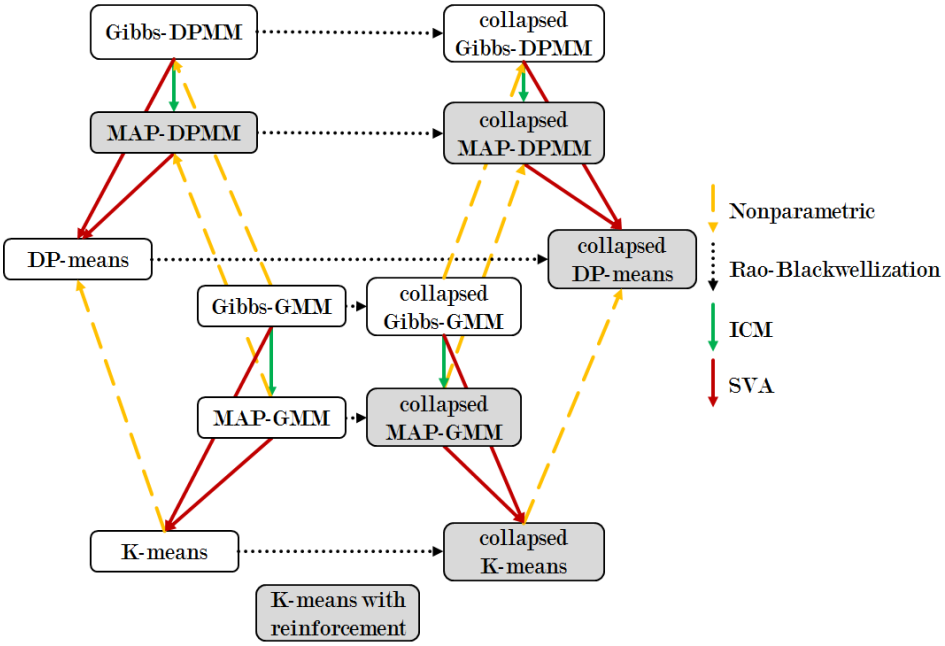


Figure 2.4: In the white boxes are well known approaches to inference in the Gaussian mixture model and the Dirichlet process mixtures. The shaded boxes are novel approaches explicitly derived and proposed in this thesis. Yellow lines point to a nonparametric extension of a parameteric algorithm; dotted lines point to a collapsed (see the text) version of an algorithm; green lines denote that the algorithm was derived using iterated conditional modes; red lines denote that the algorithm was derived using small variance asymptotics assumptions (see the text).

Each of the methods in Figure 2.4 has various trade-offs of flexibility, computational and conceptual complexity where the following chapters aim to evaluate some of those tradeo-offs. While Figure 2.4 focuses only on mixture models, similar arguments can be used to derive many other inference algorithms for more sophisticated latent variable models.

For example, we can extend our reasoning to sequential discrete latent variable models, such as the hidden Markov model (HMM) discussed in Chapter 5. The methods proposed and discussed in Chapter 5 could be summarized by a similar association chart in Figure 2.5.

[5] By collapsed construction of a model we refer to a probabilistic model with marginalized parameters.

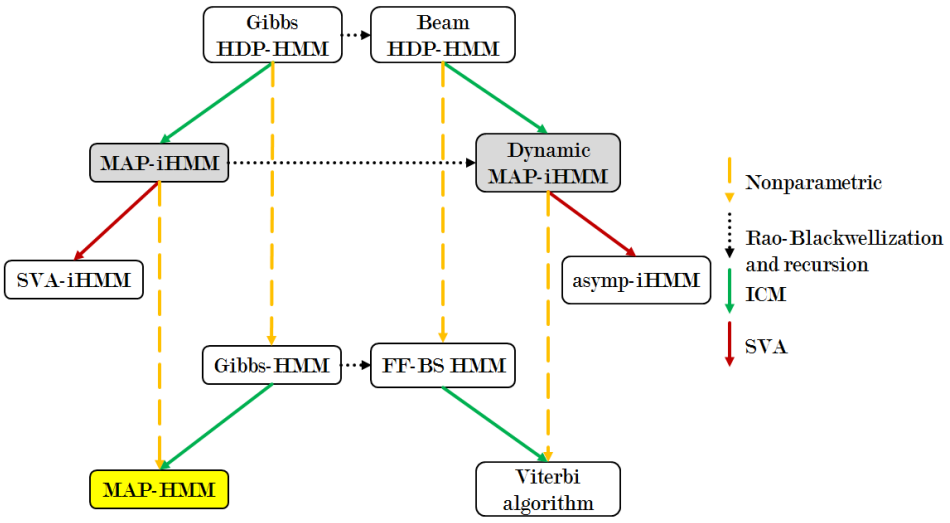


Figure 2.5: In the white boxes are well known approaches to inference in the HMMs and the nonparametric HMMs. The shaded boxes are novel approaches explicitly derived and proposed in this thesis. MAP-HMM algorithm is in a yellow box as it can be conceptualy derived based on this thesis but we have not included in the discussion due to its limited practical value. Yellow lines point to a nonparametric extension of a parameteric algorithm; dotted lines point to a collapsed (see the text) version of an algorithm; green lines denote that the algorithm was derived using iterated conditional modes; red lines denote that the algorithm was derived using small variance asymptotics assumptions (see the text).

Chapter 3

Simple deterministic inference for mixture models

3.1 Introduction

In Section 2.1, we reviewed K-means from its original geometric viewpoint. However, it can also be profitably understood from a probabilistic viewpoint, as a restricted case of the Gaussian mixture model after *small variance asymptotic* (SVA) assumptions are applied. This probabilistic framework helps us to more intuitively understand the inherent assumptions and limitations of K-means. Furthermore it allows us to easily derive simple adaptations that more accurately treat different data types by starting off with more appropriate probabilistic models and applying similar SVA assumptions. This framework will allow us to study how concepts that are well explored in Bayesian inference, such as Rao-Blackwellization, extend to deterministic clustering techniques.

As discussed in Section 2.1 K-means and SVA algorithms in general suffer from some major drawbacks. We are interested in keeping the clustering problem nearly as easy to solve as K-means, but at the same time overcoming those drawbacks. Towards that end, we propose iterative MAP methods in contrast to SVA procedures and demonstrate their flexibility and computational scalability in many scenarios; we also attempt to provide an intuition for scenarios in which they fail when compared to MCMC methods and VB inference. Motivated by infinite mixture models that adapt the number of cluster components to the data, we have focused this chapter on nonparametric clustering methods derived from DP mixtures which can also be seen as simplified methods of doing inference in those models. To the best of our knowledge, an in depth analysis of MAP and SVA methods and how Rao-Blackwellization influences their performance has not been explored in the domain of parametric models either.

Being able to adequately replace MCMC algorithms in some scenarios with simpler procedures, has the potential for reducing computational requirements involved in Bayesian modeling by orders of magnitude. This can significantly broaden the horizon of applications to which such models can be applied. Of course, this cannot always be done in a meaningful way and one might still want to use a more demanding MCMC method, but knowing where sampling can be avoided with minimum loss of information is essential in order to increase the applicability of complex Bayesian models to a wider array of real-world problems.

In this Chapter we propose a novel deterministic clustering algorithm which is more robust to initialization issues, which we call *collapsed K-means*, as well as the more rigorous and often more flexible MAP-GMM alternative. The Chapter proceeds with the motivation for using nonparametric models and then focuses

on the development of simple, *nonparametric* clustering algorithms that we call *MAP-DPMM* (Section 3.8) and *collapsed MAP-DPMM* algorithms (Raykov *et al.*, 2015b, 2016c,b). Those methods are compared to Gibbs sampling (Neal, 2000), VB inference (Blei & Jordan, 2006) and SVA methods applied to DP mixtures on synthetic and benchmark data where the true clustering of the data is known and performance can be evaluated objectively. In Section 3.10 we demonstrate the potential for MAP-DPMM algorithms applied to problems which have so far been addressed with restrictive parametric K-means. The use of a more flexible and nonparametric approach for such problems has the potential to provide more meaningful insights for the underlying structure in the data, but at the same time it keeps interpretability and scalability.

3.2 Small variance asymptotics

3.2.1 Probabilistic interpretation of K-means

Despite the substantial differences between mixture models and the K-means algorithm, there exists a well-established connection between the two, which we describe in detail here. This connection will help us better understand the assumptions inherent to geometric methods such as K-means. It will also later help us to derive efficient, fast, greedy inference methods that retain some probabilistic rigour. Let us consider a GMM where the covariance matrices of the mixture components are spherical, constant and equal, i.e. $\Sigma_k = \sigma\mathbf{I}$ for $k = 1, ..., K$, where $\mathbf{I}$ is the identity matrix. This means that (2.5) will become:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}\left(x_i \,|\mu_k, \sigma\mathbf{I}\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(x_i \,|\mu_j, \sigma\mathbf{I}\right)} = \frac{\pi_k \exp\left(-\frac{1}{2\sigma}\left\|x_i - \mu_k\right\|_2^2\right)}{\sum_{j=1}^{K} \pi_j \exp\left(-\frac{1}{2\sigma}\left\|x_i - \mu_j\right\|_2^2\right)} \tag{3.1}$$

for some constant σ. The resulting E-M algorithm still cannot increase the negative log likelihood (Equation (2.8) from Section (2.2)) at each step, but in this case it is minimizing the negative log likelihood with respect to the component means μ_k and mixture coefficients π_k only, as the covariances are fixed. Let us further consider the limit $\sigma \to 0$: as the quantity $\left\|x_i - \mu_k\right\|_2^2$ decreases, the quantity $\pi_k \exp\left(-\frac{1}{2\sigma}\left\|x_i - \mu_k\right\|_2^2\right)$ will increase and this means that at the limit the quantity in (3.1) will go to 0 most slowly for the x_i closest to the cluster centroid μ_k. The posterior probability γ_{ik} will be 0 for all x_i except for the x_i closest to the cluster centroid μ_k for which the corresponding responsibility γ_{ik} would equal to 1. It follows that at the limit, the E-step of E-M assigns an observation to its closest cluster centroid, which is the same effect as the assignment step of K-means. Further, as mentioned above, the Gaussians have fixed covariance matrices, so the M-step re-estimates only the new mean parameters. Recall that in (2.7) $\mu_1, ..., \mu_K$ are estimated as the sample means of the observations assigned to them. Also the mixing weights π_k do not influence the assignment probabilities any more, so we can safely omit this update. Therefore after taking the limit $\sigma \to 0$ the M-step is equivalent to the update step of K-means.

In summary, we have shown that if we assume fixed, identical covariance matrices across all clusters $\sigma\mathbf{I}$ and take the limit $\sigma \to 0$, for the GMM, the EM algorithm becomes equivalent to the K-means algorithm (Bishop, 2006, page 423). This is, more recently, formalized as '*small variance asymptotics*' (SVA) derivation of K-means. If we use the Bayesian mixture model as a starting point, these additional assumptions (fixed, spherical covariances which shrink to the zero matrix) have to be made for the prior parameters in order to recover the K-means algorithm.

3.2.2 K-means with reinforcement

If we use different SVA-like assumptions, motivated by the approach taken in (Roychowdhury *et al.*, 2013) (but there applied to hidden Markov models), we can derive another, novel deterministic clustering method. We start by assuming a standard Bayesian GMM. Instead of just reducing the diagonal likelihood covariance to the 0 matrix (as we did in Section (3.2.1)), let us represent the categorical distribution over the latent variables $z_1, \ldots, z_N$ in the more general exponential family form. In a standard mixture model scenario, we write the conditional probability of an indicator given the component weights as $p(z_i | \pi) = \text{Categorical}(\pi_1, \ldots, \pi_K)$. However, as the Categorical distribution is a special case of the exponential family, we can consider a more general form of that conditional by writing it in its exponential family form, with the help of the appropriate Bregman divergences:

$$p(z_i | \pi) = \exp\left(-D_\phi(z_i, \pi)\right) b_\phi(z_i) \tag{3.2}$$

where $D_\phi(\cdot)$ denotes Bregman divergence defined above in Section 2.1. The motivation for explicitly writing the likelihood of the cluster indicators in that form is to enable us to scale the variance of the distribution over z. In order for this scaling to lead to a closed form clustering algorithm, we need to assume an additional dependency (which is not part of the original graphical model from Figure 2.1) between the distribution of the cluster indicators and the component mixture distribution, in order for their diagonal variances to approach 0 simultaneously. That is, while SVA from Section 3.2 changes the underlying Bayesian GMM structure only by assuming shrinking covariance, in this framework we modify the underlying GMM such that the conditional independence of the cluster parameters and cluster indicators no longer holds. The distribution from Equation (3.2) then is replaced by a scaled one:

$$p(z_i | \pi) = \exp\left(-\hat{\xi} D_\phi(z_i, \pi)\right) b_{\tilde{\phi}}(z_i) \tag{3.3}$$

where $\hat{\xi} > 0$ is some scale factor and $\tilde{\phi} = \hat{\xi}\phi$ where the scaled distribution keeps the same expectation as (3.2). A final step before making an asymptotic argument is to assume that the component's Gaussian likelihoods are scaled by ξ for which the equality $\hat{\xi} = \lambda_1 \xi$ holds for some real λ_1. This means that when we take the limit $\xi \to \infty$, this will scale both the distribution of the categorical indicators and the component likelihoods. After taking $\xi \to \infty$ and removing the constant terms we obtain the objective function of this new SVA approach:

$$\sum_{k=1}^{K} \sum_{i:z_i=k} \left[\|x_i - \mu_k\|_2^2 + \lambda_1 D_\phi(z_i, \pi_k) \right] \tag{3.4}$$

which is optimized with respect to (z, μ, π), and where $D_\phi(z_i, \pi_k) \propto -\ln \pi_k$. We have displayed in red the additional terms that are incorporated in this new reinforced K-means, when compared to the original objective function for K-means (2.1). Optimization with respect to the mixture weights result in the empirical probability for the cluster weights $\pi_k = \frac{N_k}{N}$. So, this objective function then can be rewritten as:

$$\sum_{k=1}^{K} \left[\sum_{i:z_i=k} \|x_i - \mu_k\|_2^2 - \lambda_1 \ln \frac{N_k}{N} \right] \tag{3.5}$$

In order to optimize the modified objective for each observation x_i, we compute the K distances to each of the clusters: $\|x_i - \mu_k\|_2^2$ for $k = 1, \ldots, K$. But we also take into account the number of data points in each component by adjusting the corresponding distance for each cluster k by subtracting the $\lambda_1 \ln \frac{N_k}{N}$ term. Each

observation x_i is assigned to the closest cluster where distance is adjusted in this way. The update step of the cluster means is equivalent to the update step of the original K-means, but in addition to the centroids we now have to update the counts $N_1, \ldots, N_K$ as well.

In contrast to K-means algorithm, this new SVA approach no longer clusters the data purely based on its geometric properties. Instead, it also takes into account the number of data points in each cluster. In that respect the method has greater flexibility, but at the same time, unlike MAP-GMM, this algorithms also does not optimize the complete data likelihood of the original underlying probabilistic GMM (compare Equation (2.8) with Equation (3.5)). With the additional ad-hoc dependencies between the likelihood distribution and the distribution over the indicator variables, SVA algorithms effectively start from a different underlying 'coupled' probabilistic model which is not explicitly given. This makes those new SVA algorithms even less principled and more heuristic. While quite simple, to some extent both types of SVA methods sacrifice several key statistical principals including structural interpretability and the existence of an underlying probabilistic generative model.

3.2.3 Overview

We demonstrated how, by introducing some simplifying assumptions, deterministic methods such as K-means can be derived as inference algorithms for some associated mixture model. This is not a consequence that relies on the E-M algorithm in any way and the same SVA assumptions can be applied directly to the complete data likelihood of a mixture model to obtain objective functions like the ones in (2.1) and (3.4). If we apply the most intuitive and trivial way of optimizing each of those objectives (iteratively, one group of variables at a time), we obtain K-means like clustering methods.

In the case of Bayesian mixture models, we usually apply some additional simplifying assumptions to relax the effect of the prior and obtain neat, closed form expressions for the updates of the SVA algorithms. Consider the Gibbs sampler for a Bayesian mixture (see Section 2.3), shrinking the component covariances to the 0 matrix means that the sample updates from the posterior for each of the model parameters (including the latent variables) are replaced with deterministic updates. The same deterministic updates can be obtained if we consider the ICM inference method (see Section 2.3.4) for that Bayesian mixture model and mirror the SVA assumptions and the additional assumptions relaxing the effect of the prior. We can view ICM as a simple deterministic algorithm which optimizes the likelihood of a model, but only guaranteeing to reach a local optima where we can view the Gibbs sampler as a stochastic algorithm that optimizes the model likelihood, asymptotically to the global optimum. Applying SVA assumptions and relaxing the effect of the prior, we obtain an even simpler deterministic algorithm which locally optimizes a reduced but *degenerate* version of the original model's likelihood (in the case of GMM, the K-means objective function from 2.1). Hence, starting from a Bayesian mixture model we can think of the Gibbs sampler as a stochastic optimization algorithm for the model likelihood. The first simplification is optimizing this likelihood with the ICM method which does not make any additional model assumptions, but is deterministic and provides only locally-optimal solutions. Then, by applying further SVA assumptions to the model likelihood (which is also the objective of the ICM algorithm) we obtain the even simpler K-means like algorithms (see Figure 3.1).

Most of the arguments made for the relationship between Bayesian GMMs, ICM inference and K-means are also valid for K-means with reinforcement from Section 3.2.2, some 'coupled' Bayesian GMM model and a matching ICM method. However, we will not discuss in detail such probabilistic models as 'coupled' mixture models are unlikely to be of great interest in practice, as well as the related ICM inference procedure.

The ICM simplification step trades away any information about the uncertainty of the model fit and theoretical guarantees of global convergence, in return for significant computational and memory usage gains.

SVA methods on the other hand trade away model interpretability, and introduce degeneracy into the likelihood prohibiting us from using standard model selection, prediction analysis and out of sample computations. In addition SVA methods rely on sphericity assumptions about the model as well as equal density components (see Section 2.1 for a detailed list). In return, marginal computational efficiency is gained as well as some conceptual simplicity. For a small increase in the complexity of K-means due to additional term in the objective function, we obtain the novel K-means with reinforcement algorithm, which can at least handle differing density across clusters.

Figure 3.1: Relating the ordering of complexity of learning schemes for different Gaussian mixture model inference algorithms.

3.3 Rao-Blackwellization in mixture models

Above, we established how simple SVA and ICM clustering procedures can be derived from Gibbs sampling for the Bayesian mixture model. However, often there is more than one way to derive a Gibbs sampler and the way we do this can seriously affect the mixing time of the sampler and its computational efficiency. One of the most common strategies for optimizing the performance of a Gibbs sampler (or any MCMC method) is appropriate use of Rao-Blackwellization (see Section 2.4).

Consider the Gibbs sampler we introduced in Section (2.3.2) for inference in Bayesian mixtures. The algorithm explicitly samples the component parameters θ and the mixing parameters π despite the fact that usually we are only explicitly interested in the latent structure in the data which is conveyed by the indicator variables $z_1, \ldots, z_N$. In such scenarios, it is common to use Rao-Blackwellization (Section (2.4)) and integrate over the redundant quantities. The resulting sampler will update only the variables of explicit interest but from a different predictive posterior distribution which is more complex to compute. For such posterior predictive distributions to be analytically tractable, we need conjugacy between the prior and the likelihood of the integrated quantities, in this case θ and π. As we consider only exponential family mixture models, conjugate prior distributions for θ and π are guaranteed to exist and the resulting *compound distribution* of the indicator variables can (nearly always) be obtained analytically.

The practical downside of integrating out random variables from the probability model is that some of the conditional independence properties among the remaining variables are affected. For example, in a non-collapsed (complete form) Bayesian GMM, the likelihood of observation x_i given z_i, θ_{z_i} and π_{z_i} is independent of the rest of the data; this leads to potential for efficient parallel implementations and the trained model has a smaller representation in memory. In a collapsed representation of the Bayesian GMM the likelihood of x_i given z_i still depends on the rest of the data and so training data needs to be kept for when making predictions.

	Algorithm 3.1: Block Gibbs (spherical Gaussian)	**Algorithm 3.2**: Collapsed Gibbs (spherical Gaussian)
Input	$x_1, \ldots, x_N$: D-dimensional data K: number of clusters α: concentration parameter σ: spherical cluster variance σ_0: prior centroid variance μ_0: prior centroid variance	$x_1, \ldots, x_N$: D-dimensional data K: number of clusters α: concentration parameter σ: spherical cluster variance σ_0: prior centroid variance μ_0: prior centroid variance
Output	Posterior of indicators: $(z_1, \ldots, z_N)$ Posterior of weights: $(\pi_1, \ldots, \pi_K)$ Posterior of centroids:$(\mu_1, \ldots, \mu_K)$	Posterior of indicators: $(z_1, \ldots, z_N)$
	Sample parameters from the prior	Sample the indicators from the prior
1	$\mu_1, \ldots, \mu_K \overset{i.i.d.}{\sim} \mathcal{N}(\mu_0, \sigma_0)$ $\pi_1, \ldots, \pi_K \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$	$z_1, \ldots, z_N \sim \text{DirMulti}\left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$
2	$E_{\text{new}} = \infty$	$E_{\text{new}} = \infty$
3	**repeat**	**repeat**
4	$E_{\text{old}} = E_{\text{new}}$	$E_{\text{old}} = E_{\text{new}}$
5	**for** $i \in 1, \ldots, N$	**for** $i \in 1, \ldots, N$
6	**for** $k \in 1, \ldots, K$	**for** $k \in 1, \ldots, K$
7		$\dot{\sigma}_k^{-i} = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma} N_k^{-i}\right)^{-1}$
8		$\dot{\mu}_k^{-i} = \dot{\sigma}_k^{-i}\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\sum_{j:z_j=k, j\neq i} x_j\right)$
9		$\dot{\pi}_k^{-i} = N_k^{-i} + \alpha/K$
10	$d_{i,k} = \frac{1}{2\sigma}\|x_i - \mu_k\|_2^2 + \frac{D}{2}\ln\sigma - \ln\pi_k$	$d_{i,k} = \frac{\|x - \dot{\mu}_k^{-i}\|_2^2}{2\left(\sigma + \dot{\sigma}_k^{-i}\right)} + \frac{D}{2}\ln\left(\sigma + \dot{\sigma}_k^{-i}\right) - \ln\dot{\pi}_k^{-i}$
11	$d_{i,k} = \exp(-d_{i,k})$	$d_{i,k} = \exp(-d_{i,k})$
12	$z_i \sim \text{Categorical}\left(\frac{d_{i,1}}{\sum_k d_{i,k}}, \ldots, \frac{d_{i,K}}{\sum_k d_{i,k}}\right)$	$z_i \sim \text{DirMulti}\left(\frac{d_{i,1}}{\sum_k d_{i,k}}, \ldots, \frac{d_{i,K}}{\sum_k d_{i,k}}\right)$
13	**for** $k \in 1, \ldots, K$	
14	$\dot{\sigma}_k = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma} N_k\right)^{-1}$	
15	$\dot{\mu}_k = \dot{\sigma}_k\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\bar{x}_k\right)$	
16	$\dot{\pi}_k = \frac{N_1 + \alpha/K}{N + \alpha}$	
17	$\mu_k \sim \mathcal{N}(\dot{\mu}_k, \dot{\sigma}_k)$	
18	$(\pi_1, \ldots, \pi_K) \sim \text{Dirichlet}(\dot{\pi}_1, \ldots, \dot{\pi}_K)$	
19	$E_{\text{new}} = \sum_{k=1}^K \sum_{i:z_i=k} d_{i,k}$	$E_{\text{new}} = \sum_{k=1}^K \sum_{i:z_i=k} d_{i,k}$
20	**until** $(E_{\text{old}} - E_{\text{new}}) \to$ random	**until** $(E_{\text{old}} - E_{\text{new}}) \to$ random

The collapsed Gibbs for the special case of a Bayesian GMM with spherical covariances is contrasted with block Gibbs in Algorithm **3.1**. Note the differences in the convergence criteria (which we discussed in Section 2.3.2) of the Gibbs sampler (line 20) compared to those from Algorithms **2.1** and **2.2**. Unlike the fixed-point solution of MAP-GMM, there is no well established criteria to assess convergence of the Gibbs sampler.

We now proceed with a more general summary of collapsed Gibbs for exponential family finite Bayesian mixture models. Firstly, each z_i is sampled from:

$$p\left(z_i = k \,|z_{-i}, x\right) = \frac{\left(N_k^{-i} + \alpha/K\right) p\left(x_i \,|z_i = k\right)}{\sum_{j=1}^{K}\left(N_j^{-i} + \alpha/K\right) p\left(x_i \,|z_i = j\right)} \tag{3.6}$$

with $z_{-i} = \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N\}$ denoting all of the indicators excluding z_i. The likelihood of point i is computed by integrating out the effect of the component parameters θ from $p\left(x_i \,|\theta, z_i = k\right)$:

$$p\left(x_i \,|z_i = k\right) = \int_{\Theta} p\left(x_i \,|\theta_k, z_i = k\right) p\left(\theta\right) d\theta \tag{3.7}$$

where Θ denotes the domain of the parameters θ. In the exponential family notation used in Section 2.3, if we ignore terms independent of k the marginal likelihood of point i from cluster k is: $p\left(x_i \left|\tau_k^{-i}, \eta_k^{-i}\right.\right) \sim \exp\left[\psi_0\left(\tau_k, \eta_k\right) - \psi_0\left(\tau_k^{-i}, \eta_k^{-i}\right)\right]$ where recall that (τ_k, η_k) are the hyperparameters of the integrated natural parameters θ; superscript $-i$ stands for removing the effect of point i when estimating a quantity and $\psi_0\left(\cdot\right)$ is a normalization function automatically determined by the choice of remaining quantities. The form of $p\left(x_i \,|z\right)$ is strictly defined by the choice of likelihood and prior distribution over the parameters, where a useful list of choices for $p\left(x_i \,|\theta\right)$ and corresponding predictive likelihoods $p\left(x_i \,|z\right)$ can be found in Appendix (A). We iteratively sample the indicator values using (3.6) for each i.

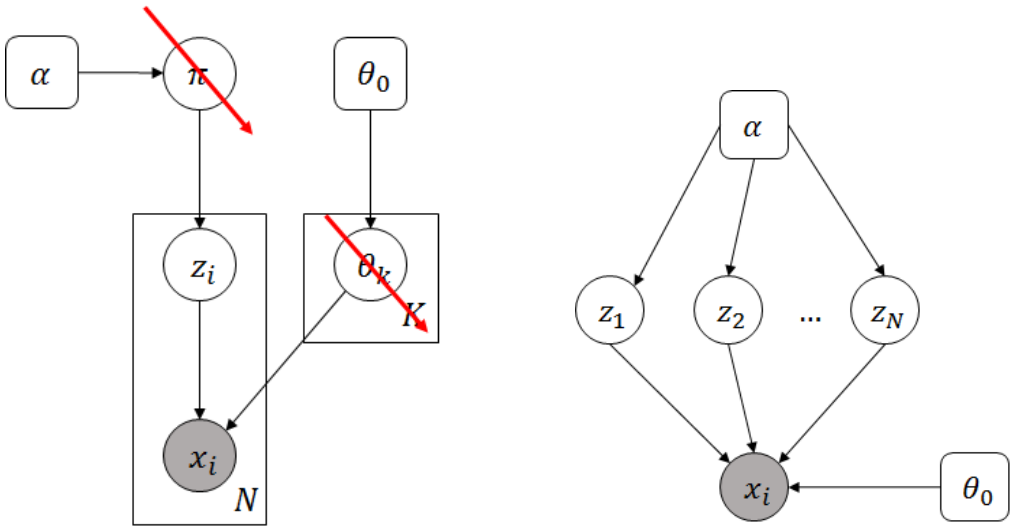


Figure 3.2: Probabilistic graphical model of the collapsed Bayesian mixture model. In the Gaussian case $\theta = \{\mu, \Sigma\}$ and $\theta_0 = \{m_0, c_0, b_0, a_0\}$. The red arrow denotes collapsing over the marked random variable.

The collapsed Gibbs sampler iterates through a reduced number of random quantities which often leads to faster mixing and more efficient inference as a result. A single iteration of collapsed Gibbs is computationally more expensive than for the block sampler (due to the broken parallelism and the more complex marginal likelihood), however this overhead is often more than compensated for with the reduced number of iterations to convergence.

Prediction and memory constraints in Rao-Blackwellization

Despite the benefits of Rao-Blackwellization, there are some cases in which its use should be carefully considered. For example, from an application perspective computational power is often not the only constrained

resource, but memory "footprint" is also important. Consider the situation where we want to do prediction and clustering of some future data, using a Bayesian GMM trained on some previous data $x_1^{train}, \ldots, x_N^{train}$. In the case of the collapsed model, we would need to keep in memory all the training data in addition to all N of the indicators z_i, to be able to make any inference about the test data. For a large amount of training data N this can be highly inefficient. By contrast, under the non-collapsed Bayesian GMM we can make predictions about future test points using only the model parameters $\left\{\theta_1^{train}, \ldots, \theta_K^{train}\right\}$ and there are only K of them rather then a multiple of N. More formally, under the non-collapsed (block) representation we use:

$$p\left(z_{N+1}^{test} \left| z^{train}, \theta, x^{train}, x^{test}\right.\right) = p\left(z_{N+1}^{test} \left| \theta, x^{test}\right.\right) \propto \begin{cases} p\left(x_{N+1}^{test} \left| \theta_1\right.\right) p\left(\theta_1\right) & \text{for } z_{N+1}^{test} = 1 \\ \vdots & \vdots \\ p\left(x_{N+1}^{test} \left| \theta_K\right.\right) p\left(\theta_K\right) & \text{for } z_{N+1}^{test} = K \end{cases} \tag{3.8}$$

for any out-of-sample calculations, where for collapsed mixture model we use the following:

$$p\left(z_{N+1}^{test} \left| z^{train}, \theta, x^{train}, x^{test}\right.\right) = p\left(z_{N+1}^{test} \left| z^{train}, x^{train}, x^{test}\right.\right) \propto \begin{cases} p\left(x_{N+1}^{test} \left| z_{j:z_j=k}\right.\right) & \text{for } z_{N+1}^{test} = 1 \\ \vdots & \vdots \\ p\left(x_{N+1}^{test} \left| z_{j:z_j=k}\right.\right) & \text{for } z_{N+1}^{test} = K \end{cases} \tag{3.9}$$

3.4 Collapsed K-means and collapsed MAP-GMM

In Section 3.3 we explored how Rao-Blackwellization can be used in probabilistic models, particularly its ability to improve computational efficiency for sampling methods. In this section we will look at the potential of collapsed models to lead to useful deterministic clustering algorithms. Mirroring the SVA derivation of K-means from Section 3.2, we can derive a new collapsed K-means from the collapsed GMM. The resulting collapsed K-means will sequentially update the assignments $z_1, \ldots, z_N$, while also keeping the cluster means μ_k^{-i} updated for each k. In collapsed K-means, cluster centroids are re-computed after any assignment changes, unlike in K-means where centroids are updated in one block after a full sweep through $z_1, \ldots, z_N$. Another difference is the way both are initialized: K-means is initialized by setting values for its cluster centroids $\mu_1, \ldots, \mu_k$; collapsed K-means is initialized by setting each of the assignments $z_1, \ldots, z_N$. We outline collapsed K-means in the Algorithm **3.3**.

Let us now consider applying the MAP approach from Section 2.3.4 to a collapsed GMM. The resulting collapsed MAP-GMM algorithm will be also sequential and we outline it in Algorithm **3.4**. In the collapsed GMM the cluster centroids $\mu_1, \ldots, \mu_K$ and the mixing coefficients $\pi_1, \ldots, \pi_K$ have been integrated out. Therefore the likelihood term (marginal likelihood) in the collapsed model $p\left(x_i \left| z_i\right.\right) = \int p\left(x_i \left| \mu, \sigma\right.\right) p\left(\mu\right) d\mu$ takes a more robust form after integrating over the uncertainty around the unknown parameter μ and taking into account the effect of $\dot{\sigma}_k^{-i}$ (the variance in the posterior of μ); additionally the MAP update of π is replaced with an update of the posterior hyperparameter $\dot{\pi}$. As with the collapsed K-means, MAP-GMM will avoid the need for initialization of the cluster centroids and also of the mixing weights; instead it will require initialization of the cluster indicators $z_1, \ldots, z_N$. In the following section, we evaluate empirically the performance of collapsed and non-collapsed K-means and MAP-GMM on synthetic data for various scenarios.

	Algorithm 3.3: Collapsed K-means		**Algorithm 3.4**: Collapsed MAP-GMM (spherical Gaussian)
Input	$x_1, \ldots, x_N$: D-dimensional data $\epsilon > 0$: convergence threshold K: number of clusters		$x_1, \ldots, x_N$: D-dimensional data $\epsilon > 0$: convergence threshold K: number of clusters α: concentration parameter $\hat{\sigma}^2$: spherical cluster variance σ_0^2: prior centroid variance
Output	$z_1, \ldots, z_N$: cluster assignments		$z_1, \ldots, z_N$: cluster assignments
1	Set z_i for all $i \in 1, \ldots, N$	**1**	Set z_i for all $i \in 1, \ldots, N$
2	$E_{\text{new}} = \infty$	**2**	$E_{\text{new}} = \infty$
3	**repeat**	**3**	**repeat**
4	$E_{\text{old}} = E_{\text{new}}$	**4**	$E_{\text{old}} = E_{\text{new}}$
5	**for** $i \in 1, \ldots, N$	**5**	**for** $i \in 1, \ldots, N$
6	**for** $k \in 1, \ldots, K$	**6**	**for** $k \in 1, \ldots, K$
7		**7**	$\dot{\sigma}_k^{-i} = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma} N_k^{-i}\right)^{-1}$
8	$\mu_k^{-i} = \frac{1}{N_k^{-i}} \sum_{j:z_j=k, j\neq i} x_j$	**8**	$\dot{\mu}_k^{-i} = \sigma_k^{-i} \left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma} \sum_{j:z_j=k, j\neq i} x_j\right)$
9		**9**	$\dot{\pi}_k = N_k^{-i} + \alpha/K$
10	$d_{i,k} = \frac{1}{2} \left\|x_i - \mu_k^{-i}\right\|_2^2$	**10**	$d_{i,k} = \frac{\left\|x - \dot{\mu}_k^{-i}\right\|_2^2}{2\left(\sigma + \dot{\sigma}_k^{-i}\right)} + \frac{D}{2} \ln\left(\sigma + \dot{\sigma}_k^{-i}\right) - \ln \dot{\pi}_k$
11	$z_i = \arg\min_{k \in 1, \ldots, K} d_{i,k}$	**11**	$z_i = \arg\min_{k \in 1, \ldots, K} d_{i,k}$
12	$E_{\text{new}} = \sum_{k=1}^{K} \sum_{i:z_i=k} d_{i,k}$	**12**	$E_{\text{new}} = \sum_{k=1}^{K} \sum_{i:z_i=k} d_{i,k}$ $- \log \Gamma(N + \alpha) - \sum_{k=1}^{K} \log \Gamma(N_k + \alpha/K)$
13	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$	**13**	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$

3.5 Comparison on synthetic data

In this section we evaluate the performance of K-means, collapsed K-means, MAP-GMM and collapsed MAP-GMM algorithms on six different synthetic Gaussian data sets with $N = 4000$ points. The data sets have been generated to demonstrate some of the non-obvious problems with the K-means algorithm.

The true clustering assignments are known so that the performance of the different algorithms can be objectively assessed. For the purpose of illustration we have generated two-dimensional data with three, visually separable clusters, to highlight specific problems that arise with K-means. To ensure that the results are stable and reproducible, we have performed 100 restarts for all of the algorithms. Collapsed K-means and collapsed MAP-GMM restarts involve a random permutation of the ordering of the data, where K-means and MAP-GMM are restarted with randomized parameter initializations.

To evaluate algorithm performance we have used *normalized mutual information* (NMI)[1] between the true and estimated partition of the data (Table 3.1), where we report average NMI scored across the 100 restarts. The NMI between two random variables is a measure of mutual dependence between them that takes values

[1] The mutual information of two discrete random variables Z and Z^* is defined as MI$= \sum_{z \in Z^*} \sum_{z \in Z} p(z, z^*) \log\left(\frac{p(z,z^*)}{p(z)p(z^*)}\right)$. To obtain a more intuitive score taking values between 0 and 1 MI is often normalized and the NMI score used instead.

Table 3.1: Comparing the clustering performance of K-means, collapsed K-means, MAP-GMM and collapsed MAP-GMM. Methods are tested on synthetic data generated from a GMM with $K = 3$, and to assess clustering performance we evaluate average NMI score and standard deviation in the brackets.

Cluster geometry	Shared cluster geometry?	Shared cluster density?	NMI K-means	NMI Collapsed K-means	NMI MAP-GMM	NMI Collapsed MAP-GMM
Spherical	No	Yes	0.64(0.11)	0.76(0.04)	0.58(0.08)	0.60(0.05)
Spherical	Yes	No	0.71(0.04)	0.73(0.01)	0.96(0.03)	0.98(0.01)
Spherical	Yes	Yes	0.49(0.32)	0.76(0.01)	0.48(0.32)	0.92(0.04)
Elliptical	No	Yes	0.58(0.13)	0.53(0.11)	0.56(0.10)	0.56(0.08)
Elliptical	No	No	0.95(0.15)	0.98(0.01)	0.95(0.13)	0.99(0.01)
Elliptical	No	No	0.56(0.02)	0.59(0.1)	0.56(0.02)	0.60(0.02)

Table 3.2: Number of iterations to convergence of K-means, collapsed K-means, MAP-GMM and collapsed MAP-GMM with standard deviation in the brackets. The computational cost per iteration is not exactly the same for different algorithms, but it is comparable.

Convergence K-means	Convergence Collapsed K-means	Convergence MAP-GMM	Convergence Collapsed MAP-GMM
17(8.5)	2(0)	21(11.7)	2(0)
48(21.3)	2(0)	9(2.7)	2(0)
7(3.6)	2(0)	7(3.3)	2(0)
28(14.2)	2(0)	30(18)	2(0)
7(4)	2(0)	8(7)	2(0)
33(10)	2(0)	34(11)	2(0)

between 0 and 1 where the higher score means stronger dependence. NMI scores close to 1 indicate good agreement between the estimated and true clustering of the data.

We also report the average number of iterations to convergence of each algorithm in Table 3.2 as an indication of the relative computational cost involved, where the iterations include only a single run across all data and parameters of the corresponding algorithm.

From Table 3.1 we observe that both MAP and SVA methods derived from collapsed models are more robust to initialization and their performance is influenced less by random restarts. They also converge significantly faster, although we have to take into account that their iterations are more expensive due to the coupling introduced by the Rao-Blackwellization (see Figure 3.2). A single iteration of K-means sweeps through N observations, computing K distances for each; after finishing the full sweep through the observations all K cluster centroids are updated. At the same time an iteration of the collapsed K-means involves sweeping through N observations, computing K distances for each and also updating K cluster means. Using caching we can substantially speed up the updates of the means in the collapsed K-means.

MAP methods handle better violations of the inherent assumption of shared cluster density of K-means (see from Section 2.1), because they incorporate reinforcement terms. We observe that using the wrong values for a fixed component covariance may often be worse than simply shrinking its value to the 0 matrix, as K-means performs slightly better than MAP on elliptical data. However, unlike SVA algorithms, it is straightforward to extend MAP methods to non-spherical data (with unknown component covariances) using Bayesian GMM with NIW priors on the component parameters. In fact, such an extension practically always outperforms K-means (see (Raykov *et al.*, 2016c)).

Despite the tremendous popularity of K-means and the potential of the other techniques presented and

reviewed in this section, they all suffer from some common drawbacks which occur due to the construction of the underlying probabilistic model (the GMM). The parametric methods here require fixing the number of components K a-priori and further keeping K the same unless the method is completely retrained. This constraint will lead to less robust clustering procedures which are too sensitive to statistical outliers in the data and which fail to handle even small dissimilarities between the underlying model we assume and the observed data. We reviewed in detail some existing methods which use regularization to choose K in Chapter 1, but regularization simply does not provide rigorous statistical machinery to handle such issues.

In the next sections we turn to DP mixture models and using SVA and MAP reasoning we derive some novel, principled and yet more flexible nonparametric clustering algorithms.

3.6 Nonparametric clustering alternatives

In the sections above, we demonstrated how meaningful clustering algorithms can be derived by applying SVA and MAP inference to collapsed and non-collapsed finite mixture models. Here we will study how more flexible, infinite mixture models (also known as the Dirichlet process mixture model (DPMM)) can be designed and in later sections we will focus on the new, nonparametric clustering procedures we can derive by applying Gibbs inference, SVA inference and iterative MAP inference to the DPMM.

3.6.1 Gibbs sampling for DPMM

Inference for a given probabilistic model using Gibbs sampling consists of an iterative sweep through the conditionals for each random variable in the model and sampling its value, one at a time or in batches. In the case of finite mixture models (see Section 2.3.2) typically the conditionals for each random variable are available in closed form as can be clearly seen in Section 2.3 for any exponential family mixture model. However, in the DPMM the number of components modeling the data is not fixed, as we take $K \to \infty$. This means we can no longer sample $\pi_1, \ldots, \pi_K, \ldots$ in a straightforward way. In order to keep the data likelihood tractable, we often integrate out the infinite dimensional mixing parameters π from the model (Neal, 2000; Rasmussen, 1999). The integration can be performed analytically as the DP is conjugate to the categorical likelihood of the indicators and leads to CRP modeling $z_1, \ldots, z_N$. We proceed by reviewing two widely-used Gibbs sampling algorithms based on the CRP construction of the DPMM. The two algorithms are very similar (Algorithm 1 and 3 from (Neal, 2000)) and differ only in their treatment of the component parameters $\theta_1, \ldots, \theta_K$. Both samplers integrate over π, but the collapsed CRP-based Gibbs sampler (collapsed Gibbs-DPMM) in Section 3.6.1 integrates over all component parameters θ as well and the sampler (Gibbs-DPMM) in Section 3.6.1 explicitly refers to the all represented component parameters $\theta_1, \ldots, \theta_K$.

CRP-based Gibbs sampler (Gibbs-DPMM)

If we collapse over the mixing parameters in an exponential family DPMM, implicitly we are invoking the corresponding CRP mixture model from Section 2.5.4:

$$\begin{aligned} (z_1, \ldots, z_N) &\sim \text{DirMulti}\,(\alpha) \\ \theta_k &\sim G_0 \\ x_i &\sim F\,(\theta_{z_i}) \end{aligned} \qquad \overset{K\to\infty}{} \qquad \begin{aligned} (z_1, \ldots, z_N) &\sim \text{CRP}\,(N_0, N) \\ \theta_k &\sim G_0 \\ x_i &\sim F\,(\theta_{z_i}) \end{aligned} \tag{3.10}$$

On the left is the matching generative model for a finite mixture whenever we assume K is finite and fixed. The Gibbs sampler for the CRP mixture will iterate between updates for the indicator variables z and the

parameters θ until convergence. For each point i we compute:

$$p\left(z_i = k \,|z_{-i}, x, \theta\right) = \begin{cases} \frac{N_k^{-i} p(x_i|\theta_k)}{\sum_{k=1}^{K} N_k^{-i} p(x_i|\theta_k) + N_0 p(x_i|\tau_0,\eta_0)} & \text{for existing } k = 1, \ldots, K \\ \frac{N_0 p(x_i|\tau_0,\eta_0)}{\sum_{k=1}^{K} N_k^{-i} p(x_i|\theta_k) + N_0 p(x_i|\tau_0,\eta_0)} & \text{for new } k = K+1 \end{cases} \tag{3.11}$$

where $p\left(x_i \,|\theta_k\right)$ and $p\left(x_i \,|\tau_0, \eta_0\right)$ are respectively the exponential family likelihood of point i given parameter θ_k and the predictive likelihood of point i being drawn from the prior. The predictive likelihood is computed by integrating θ, $p\left(x_i \,|\tau_0, \eta_0\right) = \int p\left(x_i \,|\theta\right) p\left(\theta\right) d\theta$ and the likelihood term $p\left(x_i \,|\theta_k\right) = \exp\left(\left\langle g\left(x_i\right), \theta_{z_i}\right\rangle - \psi\left(\theta_{z_i}\right) - h\left(x_i\right)\right)$ is computed in the same way as in (2.11) from Section 2.3.2 for finite mixture models. The motivation behind using $p\left(x_i \,|\tau_0, \eta_0\right)$ for unrepresented components is that, a priori, there are infinitely many unrepresented θ_0 from which a point can be sampled (all of which have diminishing support). Therefore, we can obtain a closed form expression for the probability of a single point under the prior only by integrating out the infinite dimensional parameter space. When a new value $K+1$ is sampled for an indicator z_i, we proceed by sampling a new component parameter θ_{K+1} from the posterior for a single point cluster; $p\left(\theta_{K+1} \,|\tau_{K+1}, \eta_{K+1}\right)$ with $\tau_{K+1} = \tau_0 + g\left(x_1\right)$ and $\eta_{K+1} = \eta_0 + 1$.

For each k we update component parameter θ_k from the exponential family posterior given all points assigned to k:

$$p\left(\theta_k \,|\tau_k, \eta_k\right) = \exp\left(\left\langle \theta_k, \tau_k\right\rangle - \eta_k \psi\left(\theta_k\right) - \psi_0\left(\tau_k, \eta_k\right)\right) \tag{3.12}$$

with hyperparameters τ_k and η_k being sufficient statistics of data associated with k: $\tau_k = \tau_0 + \sum_{j:z_j=k} g\left(x_j\right)$ and $\eta_k = \eta_0 + N_k$.

Collapsed CRP-based Gibbs sampler (collapsed Gibbs-DPMM)

From the Rao-Blackwell theorem in Section 2.4, the minimum variance estimation of the cluster indicators of a DPMM is obtained after marginalizing out the remaining unknown quantities θ from their joint statistic. The cluster indicators parametrize the partition implicit in the data and so they are going to be the quantity of explicit interest when we are learning the clustering of the data. The generative model for a collapsed DPMM is:

$$\begin{aligned} \left(z_1, \ldots, z_N\right) \sim\ & \text{CRP}\left(N_0, N\right) \\ x_i \sim & F^{-i}\left(\tau_{z_i}^{-i}, \eta_{z_i}^{-i}\right) \end{aligned} \tag{3.13}$$

where F^{-i} denotes the marginal likelihood distribution which arises from the Rao-Blackewellization of θ and $\left(\tau_k^{-i}, \eta_k^{-i}\right)$ are the posterior hyperparameters for component k computed after removing the effect of point i on that component. The corresponding collapsed Gibbs-DPMM iterates between updates for each of the indicators z_i while holding the rest of the indicators $z_{-i} = z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N$ fixed. For each observation, we compute:

$$p\left(z_i = k \,|z_{-i}, x\right) = \begin{cases} \frac{N_k^{-i} p\left(x_i|\tau_k^{-i}, \eta_k^{-i}\right)}{\sum_{k=1}^{K} N_k^{-i} p\left(x_i|\tau_k^{-i}, \eta_k^{-i}\right) + N_0 p(x_i|\tau_0,\eta_0)} & \text{for existing } k = 1, \ldots, K \\ \frac{N_0 p(x_i|\tau_0,\eta_0)}{\sum_{k=1}^{K} N_k^{-i} p\left(x_i|\tau_k^{-i}, \eta_k^{-i}\right) + N_0 p(x_i|\tau_0,\eta_0)} & \text{for new } k = K+1 \end{cases} \tag{3.14}$$

As the component parameters have been integrated out, when a new value $K+1$ is chosen, we only need to increase the number of represented clusters. The sampler iterates through the updates for the different z_i's until convergence is reached. Examples of posterior predictive densities for different exponential family likelihoods are presented in Appendix A and those can be directly substituted in both samplers.

3.7 Deterministic inference for Dirichlet process mixtures

The execution time of MCMC inference for the DPMM is usually larger than MCMC inference for finite models and this can seriously restrict the practical value of nonparametric mixtures. A widely-used alternative to MCMC inference in latent variable models is to use deterministic variational Bayes (VB) techniques and some VB methods for inference in DPMMs are briefly discussed below. Another deterministic method for inference in DPMMs was described in Daumé III (2007). This approach is based on a combinatorial search that is guaranteed to find the optimum for objective functions which have a specific computational tractability property. As the DPMM complete data likelihood does not have this particular tractability property, the algorithm from Daumé III (2007) is only approximate for the DPMM, and also sample-order dependent. On the other hand, Dahl *et al.* (2009) described an algorithm that is guaranteed to find the global optimum in $N(N+1)$ computations, but only in the case of *univariate product partition models*[2] with non-overlapping components. While the CRP mixture model with exponential family components falls in this class of product partition models, the restrictions that the model should be univariate and underlying components should be non-overlapping hardly ever holds in practice.

Wang & Dunson (2011) presented another approach for fast inference in DPMMs which discards the exchangeability assumption of the data partitioning and instead assumes the data is in the correct ordering. Then a greedy, repeated "uniform resequencing" is proposed to maximize a pseudo-likelihood that approximates the DPMM complete data likelihood. However, this method does not have any guarantees for convergence even to a local optima.

Our novel approach, introduced below, does not make any further assumptions beyond the model structure, and being derived from the Gibbs sampler does not suffer from sample-order dependency and is guaranteed to converge to at least a local optima.

3.7.1 Variational inference for DPMMs

Blei & Jordan (2006) first introduced VB inference for the DPMM (VB-DPMM), but it involves truncating the variational distribution of the joint DPMM posterior. Later, a related collapsed variational method was proposed in (Teh *et al.*, 2007) which reduces the inevitable truncation error by working on a reduced-dimensional parameter space, but collapsed VB methods are based on a sophisticated family of marginal likelihood bounds for which optimization is challenging. *Streaming variational* methods (Broderick *et al.*, 2013b) obtain significant computational speed up by optimizing local variational bounds on batches of data visiting data points only once, but as a result they can easily become trapped at a poor fixed point. Similarly, *stochastic variational* methods (Wang *et al.*, 2011) also allow for a single pass through the data, but sensitivity to initial conditions increases substantially. Alternatively, methods which learn *memoized* statistics[3] of the data in a single pass (Hughes & Sudderth, 2013; Hughes *et al.*, 2015a), have recently shown promise.

Here we briefly summarize the benchmark VB inference algorithm for the DPMM introduced in (Blei & Jordan, 2006) for exponential family data (VB-DPMM for short). Recall that in mean-field variational methods we aim to optimize the KL divergence with respect to some variational distribution which for the DP mixture we will denote with $q(\beta, \theta, z)$. In the stick-breaking representation of DP mixtures of exponential family, β are the stick lengths, θ are the component parameters and z are the cluster indicators. We are interested in minimizing the KL divergence between the family of distributions $q(\beta, \theta, z)$ and the true posterior

[2]Product partition models are a class of probability models parametrized by a set partition. Such models imply that items in different partition components are independent.

[3]The memoized statistics updates speed up the expensive to compute standard sufficient statistics updates by storing the repeated sufficient statistics values and returning cached results whenever the same update is needed.

of the latent variables and the model parameters $p(\beta, \theta, z \,|x, \theta_0, N_0)$:

$$\mathrm{KL}\left(q(\beta,\theta,z)\,\|p(\beta,\theta,z\,|x,\theta_0,N_0)\right) = \mathbb{E}_q\left[\ln q(\beta,\theta,z)\right] - \mathbb{E}_q\left[\ln p(\beta,\theta,z,x\,|\theta_0,N_0)\right] + \ln p(x\,|\theta_0,N_0) \quad (3.15)$$

where we can safely omit the dependence on the component hyperparameters θ_0 and the concentration parameter N_0 as in the whole thesis these are assumed to be fixed quantities. The expectations $\mathbb{E}_q(\cdot)$ are all evaluated with respect to the appropriate variational distribution (e.g. the random variable described by the corresponding distribution). The minimization of the KL divergence is often cast alternatively (see section 2.3.3) as maximization of a lower bound on the log marginal likelihood:

$$\ln p(x) \geq \mathbb{E}_q\left[\ln p(\beta,\theta,z,x)\right] - \mathbb{E}_q\left[\ln q(\beta,\theta,z)\right] \quad (3.16)$$

For the stick-breaking construction of the DPMM this bound can be re-written as:

$$\begin{aligned} \ln p(x) \geq \quad & \mathbb{E}_q\left[\ln p(\beta)\right] + \mathbb{E}_q\left[\ln p(\theta)\right] \\ & + \sum_{i=1}^{N}\left(\mathbb{E}_q\left[\ln p(z_i\,|\beta)\right] + \mathbb{E}_q\left[\ln p(x_i\,|z_i)\right]\right) \\ & - \mathbb{E}_q\left[\ln q(\beta,\theta,z)\right] \end{aligned} \quad (3.17)$$

In order to obtain tractable optimization, a restricted family of variational distributions $q(\beta, \theta, z)$ shall be considered, where in (Blei & Jordan, 2006) fully factorized variational distributions are assumed. Furthermore, we must find a family of distributions which approximates the infinite-dimensional parameters space expressed in terms of the sequences $\beta = (\beta_k)_{k=1}^{\infty}$ and $\theta = (\theta_k)_{k=1}^{\infty}$. (Blei & Jordan, 2006) suggested using truncated variational distributions as the stick-breaking construction refers to possibly infinite number of β and θ. We model the exact distributions of the infinite dimensional $\beta = (\beta_k)_{k=1}^{\infty}$ and $\theta = (\theta_k)_{k=1}^{\infty}$ with a truncated number of L variational distributions where L is finite and fixed. Incorporating this truncation and the assumption of fully factorized variational distributions, we consider a family of variational distributions that take the form:

$$q(\beta,\theta,z) = \prod_{k=1}^{L-1} q_{\gamma_k}(\beta_k) \prod_{k=1}^{L} q_{\tau_k}(\theta_k) \prod_{i=1}^{N} q_{\phi_i}(z_i) \quad (3.18)$$

where each $q_{\gamma_k}(\beta_k)$ are beta distributions; each $q_{\tau_k}(\theta_k)$ are exponential family distributions with natural parameters τ_k; $q_{\phi_i}(z_i)$ are categorical distributions and $(\gamma_1, \ldots, \gamma_{L-1}, \tau_1 \ldots, \tau_L, \phi_1, \ldots, \phi_N)$ are the parameters of the variational distributions which we need to infer (learn the parameter values which maximize the lower bound in (3.17)).

We proceed now with a short summary of a *coordinate ascent* algorithm which optimizes the bound in (3.17) with respect to the variational parameters. Most of the terms can be directly obtained using the fact that the likelihood is from an exponential family and the model priors are conjugate. From the truncation assumption made earlier the expectation $\mathbb{E}_q\left[\ln p(z_i\,|\beta)\right]$ can be re-written as:

$$\mathbb{E}_q\left[\ln p(z_i\,|\beta)\right] = \sum_{k=1}^{T} q(z_i > t)\,\mathbb{E}_q\left[\ln(1-\beta_k)\right] + q(z_i = k)\,\mathbb{E}_q\left[\ln \beta_k\right] \quad (3.19)$$

with

$$\begin{aligned} q(z_i = k) &= \phi_{i,k} \\ q(z_i > k) &= \sum_{j=t+1}^{T} \phi_{i,k} \\ \mathbb{E}_q[\ln V_k] &= \Psi(\gamma_{i,1}) - \Psi(\gamma_{i,1} + \gamma_{i,2}) \\ \mathbb{E}_q[\ln(1 - V_k)] &= \Psi(\gamma_{i,2}) - \Psi(\gamma_{i,1} + \gamma_{i,2}) \end{aligned} \tag{3.20}$$

with $\Psi(\cdot)$ denoting the digamma function. Using this expression for $\mathbb{E}_q[\ln p(z_i|\beta)]$ and optimizing the rest of the terms in the bound from (3.17) we get a mean-field coordinate ascent which yields the following updates for the variational parameters:

$$\begin{aligned} \gamma_{k,1} &= 1 + \sum_{i=1}^{N} \phi_{i,k} \\ \gamma_{k,2} &= \alpha + \sum_{i=1}^{N} \sum_{j=k+1}^{T} \phi_{i,j} \\ \tau_{k,1} &= \lambda_1 + \sum_{i=1}^{N} \phi_{i,t} x_i \\ \tau_{k,2} &= \lambda_2 + \sum_{i=1}^{N} \phi_{i,k} \\ \phi_{i,k} &\propto \exp(S_t) \end{aligned} \tag{3.21}$$

for k indexing the first T components $k \in \{1, \ldots, T\}$ and $i \in \{1, \ldots, N\}$; we have used S_t to denote the expression:

$$S_t = \mathbb{E}_q[\ln V_k] + \sum_{i=1}^{k-1} \mathbb{E}_q[\ln(1 - V_i)] + \mathbb{E}_q[\theta_k]^T x_n - \mathbb{E}_q[\psi(\theta_k)] \tag{3.22}$$

where recall that $\psi(\cdot)$ denotes the log partition factor determined from the choice of exponential family distribution we use to model the data. Iterating the updates in (3.21) optimizes the variational bound on the log marginal probability from (3.17) with respect to the variational parameters. For a more detailed derivation of the updates in (3.21), we refer the reader to (Blei & Jordan, 2006).

Updating at each iteration the corresponding single parameters amounts to performing coordinate ascent in the KL divergence. To summarize, in Gibbs sampling the random variables in the model are sampled one at a time from the corresponding posterior of that variable; when doing iterative MAP inference, the random variables are updated by taking the values that maximize their corresponding posterior; in mean-field VB we update some approximate variational parameters by setting their values so that they maximize the lower bound of the marginal likelihood.

3.7.2 Small variance asymptotics methods for DPMMs

SVA for DPMM

Starting from the CRP-based Gibbs sampler described in Section 3.6.1 with some simplifying assumptions, Jiang *et al.* (2013) describes the use of SVA reasoning to arrive at a deterministic inference algorithm for the exponential family DPMM. Consider the DPMM (3.10), but with a scaled exponential family likelihood $\tilde{F}\left(\tilde{\theta}\right)$ that is parametrized by a scaled natural parameter $\tilde{\theta} = \xi\theta$ and the log-partition function $\tilde{\psi}\left(\tilde{\theta}\right) = \xi\psi\left(\tilde{\theta}/\xi\right)$ for some $\xi > 0$. Further assume that the prior parameters of the natural parameter are also scaled appropriately,

such that $\tilde{\tau} = \frac{\tau}{\xi}$ and $\tilde{\eta} = \frac{\eta}{\xi}$. It is straightforward to see that the conjugate prior of $\tilde{\psi}$ will be also scaled and so $\tilde{\phi} = \xi\phi$. Then $\tilde{F}\left(\tilde{\theta}\right)$ has the same mean as $F(\theta)$, but with scaled covariance, $\text{cov}\left(\tilde{\theta}\right) = \text{cov}(\theta)/\xi$. Let us also assume that the concentration parameter of the DPN_0 is a function of ξ, η and τ, taking the form:

$$N_0 = \left(g_{\tilde{\phi}}\left(\frac{\tau}{\xi}, \frac{\eta}{\xi}\right) \left(\frac{2\pi}{\xi+\eta}\right)^{D/2} \xi^D \right)^{-1} \exp(-\xi\lambda) \tag{3.23}$$

for some free parameter λ that will replace the concentration parameter in the new formulation; D denotes dimension of the data and is unrelated to the later $D_\phi(\cdot)$. Following (Jiang *et al.*, 2013) we can write out the Gibbs sampler probabilities in terms of Bregman divergences $D_\phi(\cdot)$ after canceling out $f_{\tilde{\phi}}(x_i)$ terms from all probabilities:

$$p(z_i = k \,|z_{-i}, x_i, \xi, \mu) = \frac{N_{k,-i} \exp(-\xi D_\phi(x_i, \mu_k))}{C_{x_i} \exp(-\xi\lambda) + \sum_{j=1}^{K} N_j \exp(-\xi D_\phi(x_i, \mu_j))}$$
$$p(z_i = K+1 \,|z_{-i}, x_i, \xi, \mu) = \frac{C_{x_i} \exp(-\xi\lambda)}{C_{x_i} \exp(-\xi\lambda) + \sum_{j=1}^{K} N_j \exp(-\xi D_\phi(x_i, \mu_j))}$$

where C_{x_i} approaches a positive, finite constant for a given x_i as $\xi \to \infty$ and we have used the fact that for a Bregman divergence, $D_{\xi\phi}(\cdot,\cdot) = \xi D_\phi(\cdot,\cdot)$. Now as we take the limit $\xi \to \infty$, the above probabilities will become 0 for all $k \in \{1, \ldots, K+1\}$ except for k leading to the smallest value from the set $\left\{ \{D_\phi(x_i, \mu_k)\}_{k=1}^{K}, \lambda \right\}$ for which the probability approaches 1. That is, the data point x_i will be assigned to the nearest cluster with Bregman divergence at most λ. If the closest mean has a divergence greater than λ, we create a new cluster containing only x_i.

At the limit, the posterior distribution over the cluster parameters for some component k is concentrated around the sample mean of points assigned to that component , $\frac{1}{N_k}\sum_{i=1}^{N_k} x_i$. The resulting algorithm attempts to minimize the following objective function with respect to (z, μ):

$$\sum_{k=1}^{K} \sum_{i:z_i=k} D_\phi(x_i, \mu_k) + \lambda K \tag{3.24}$$

The same objective function was utilized in (Banerjee *et al.*, 2005) in the context of finite mixture models.

Although this algorithm is straightforward, it has various drawbacks in practice. The most troublesome is that the functional dependency between the concentration parameter and the covariances destroys the reinforcement (*rich-get-richer*) property of the DPMM because the counts of assignments to components $N_{k,-i}$ no longer influence which component gets assigned to an observed data point. Only the geometry in the data space matters: a new cluster is created by comparing the parameter λ against the distances between cluster centers and data points so that the number of clusters is controlled by the geometry alone, and not by the number of data points already assigned to each cluster. For high-dimensional datasets, it is not clear how to choose the parameter λ. By contrast, in the DPMM Gibbs sampler, the concentration parameter N_0 controls the rate at which new clusters are produced in a way which is independent of the geometry. Another problem that arises is that the component likelihood distributions collapse to point mass Dirac delta distributions. The Dirac delta component likelihoods lead to a degenerate model likelihood which becomes infinite and causes model selection techniques (such as marginal likelihood comparison) to be meaningless. For example, we cannot choose parameters such as λ using standard model selection methods (for example cross-validation).

SVA for DPMM with reinforcement

(Jiang *et al.*, 2013) mirrors the K-means derivation from GMM and extends it to the exponential family DPMM. However, in Section 3.2.2 we demonstrated that SVA reasoning can lead to more than one deterministic inference algorithm for a probabilistic model. Herein we mirror the approach from Section 3.2.2 in the context of the DPMM and derive a nonparametric SVA algorithm with reinforcement. Instead of just reducing the diagonal likelihood covariance to the 0 matrix, we represent the categorical distribution over the latent variables $z_1, \ldots, z_N$ in the more general exponential family form. The conditional distribution of the cluster indicator for point i given the mixture weights is given by:

$$p\left(z_i \left| \pi \right.\right) = \exp\left(-D_\phi\left(z_i, \pi\right)\right) b_\phi\left(z_i\right) \tag{3.25}$$

where $\pi = (\pi_k)_{k=1}^K$ is the vector of mixture weights. Written in this form now we can also scale the variance of the categorical distribution over z. Let us replace the distribution (3.25) by a scaled one:

$$p\left(z_i \left| \pi \right.\right) = \exp\left(-\hat{\xi} D_\phi\left(z_i, \pi\right)\right) b_{\tilde{\phi}}\left(z_i\right) \tag{3.26}$$

with $\tilde{\phi} = \hat{\xi}\phi$. The scaled distribution in (3.26) will keep the same mean as the distribution in (3.25). We assume that the likelihood $\tilde{F}\left(\tilde{\theta}\right)$ is scaled with ξ for which the equality $\hat{\xi} = \lambda_1 \xi$ holds for some real λ_1 (to obtain later closed form updates). Now, taking $\xi \to \infty$ would result in the appropriate scaling. After taking the limit and removing the constant terms we obtain the objective function of this new SVA approach:

$$\sum_{k=1}^{K} \sum_{i:z_i=k} D_\phi\left(x_i, \mu_k\right) + \lambda_1 D_\phi\left(z_i, \pi_k\right) + \lambda K \tag{3.27}$$

which is optimized with respect to (z, μ, π), and where $D_\phi\left(z_i, \pi_k\right) \propto -\ln \pi_k$. Optimization with respect to the mixture weights results in the empirical probability for the cluster weights $\pi_k = \frac{N_k}{N}$. So, this objective function then can be rewritten as:

$$\sum_{k=1}^{K} \sum_{i:z_i=k} D_\phi\left(x_i, \mu_k\right) - \lambda_1 \ln \frac{N_k}{N} + \lambda K \tag{3.28}$$

The E-M procedure that tries to optimize this objective function computes, for each observation x_i, the K divergences to each of the existing clusters: $D_\phi\left(x_i, \mu_k\right)$ for $k = 1, \ldots, K$. Then, it takes into account the number of data points in each component by adjusting the corresponding divergence with subtraction of term $\lambda_1 \ln \frac{N_k}{N}$ (for each k). After computing these adjusted distances, observation x_i is assigned to the closest cluster unless λ is smaller than all of these adjusted distances, in which case a new cluster is created. The cluster means are updated with the sample mean of observations assigned to each cluster, and in addition we now have to update the counts $N_1, \ldots, N_K$.

In contrast to the SVA algorithm proposed by (Jiang *et al.*, 2013), this novel SVA algorithm with reinforcement no longer clusters the data purely based on its geometric properties, but also takes into account the number of data points in each cluster. In that respect this SVA method has greater flexibility, but at the same time, unlike MAP-DP, we can see that none of the SVA algorithms actually optimize the complete data likelihood of the original underlying DPMM. Both SVA methods modify to some extent the original probabilistic model and so, while being quite simple, they sacrifice several key statistical principals including structural interpretability and the existence of a principled probabilistic generative model.

The DP-means algorithm

Due to its close relationship to K-means, let us consider a special case of the SVA method from (Jiang *et al.*, 2013). Consider the case of a DPMM with Gaussian components, then with similar SVA assumptions (Kulis & Jordan, 2011) derive a non-parametric version of the K-means algorithm, referred to as the *DP-means* algorithm. Assume each Gaussian component in a DPMM is spherical with identical covariance $\Sigma_k = \sigma\mathbf{I}$, and the variance parameter $\sigma > 0$ is assumed known and hence fixed in the algorithm. To obtain simple, closed form updates Kulis & Jordan assume a zero mean Gaussian prior with covariance $\rho\mathbf{I}$ and fixed $\rho > 0$ over the cluster means. Again a functional dependency between the concentration parameter N_0 and the covariances is assumed, which is $N_0 = \sqrt{1+\frac{\rho}{\sigma}} \cdot \exp\left(-\frac{\lambda}{2\sigma}\right)$, for some new parameter $\lambda > 0$. The probability of assigning observation i to cluster k can then be expressed as:

$$p\left(z_i = k \,|\mu_k, \Sigma_k\right) \propto N_k^{-i} \exp\left(-\frac{1}{2\sigma}\left\|x_i - \mu_k\right\|_2^2\right) \tag{3.29}$$

and the probability for creating a new cluster is:

$$p\left(z_i = K+1 \,|G_0\right) \propto \exp\left(-\frac{1}{2\sigma}\left[\lambda + \frac{\sigma}{\rho+\sigma}\left\|x_i\right\|_2^2\right]\right) \tag{3.30}$$

In the SVA limit $\sigma \to 0$ the probability over $z_i = k$ collapses to 1 when μ_k has the smallest distance to x_i; or instead, the probability of creating a new cluster becomes 1 when λ is smaller than any of these distances. A new cluster is created if there are any observed data points for which λ is smaller than the distance from that data point to any existing component mean vector. If a new component is generated, it will have $\mu_{k+1} = x_i$ because in the limit, the covariance of the posterior over μ_{k+1} becomes zero.

The component parameter update stage simplifies to the K-means update, i.e. the means of each component are simply replaced by the mean of every observation assigned to that component. This occurs because by conjugacy the posterior over the component means is multivariate Gaussian and as $\sigma \to 0$ the likelihood term dominates over the prior. See in Algorithm **3.5**.

3.8 Iterative maximum-a-posteriori inference

Similar to the treatment of finite mixture models, if we start from the DPMM, depending upon whether we decide to integrate out certain variables or not, we can obtain statistical models with different forms that will lead to different iterative MAP algorithms. Various constructions of the DPMM have been explored and in this Section we present two closely related iterative MAP clustering algorithms which are derived from the CRP mixture model in (3.10) (*MAP-DPMM* algorithm) and from the collapsed construction of DPMM (with collapsed component parameters) in (3.13) (*collapsed MAP-DPMM* algorithm).

3.8.1 Collapsed MAP-DPMM algorithm

In this section we propose a novel DPMM inference algorithm[4] based on iteratively updating the cluster indicators with the values that maximize their posterior (MAP values). The cluster parameters are integrated out. This algorithm can also be seen as an an "exact" version of the *maximization-expectation* (M-E) algorithm presented in (Welling & Kurihara, 2006). It is exact in the following sense: while the M-E algorithm is a kind of VB and makes a factorization assumption which departs from the underlying probabilistic

[4]This collapsed MAP-DPMM algorithm has been previously published in Raykov *et al.* (2015b,a, 2016b,c). In those peer-reviewed publications we call the collapsed MAP-DPMM algorithm "MAP-DP" for short.

model, our algorithm is derived directly from the collapsed Gibbs-DPMM. Therefore, our algorithm does not introduce or require an assumption that the joint distribution over the model parameters and the latent variables factorizes into independent factors. The essential idea is to use conditional modal point estimates rather than samples from the conditional probabilities used in Gibbs sampling.

	Algorithm 3.5: DP-means		**Algorithm 3.6**: Collapsed MAP-DPMM (spherical Gaussian)
Input	$x_1, \ldots, x_N$: D-dimensional data $\epsilon > 0$: convergence threshold λ: new cluster threshold		$x_1, \ldots, x_N$: D-dimensional data $\epsilon > 0$: convergence threshold N_0: prior count $\hat{\sigma}^2$: spherical cluster variance σ_0^2: prior centroid variance μ_0: prior centroid location
Output	$z_1, \ldots, z_N$: cluster assignments $\mu_1, \ldots, \mu_K$: cluster centroids K: number of clusters		$z_1, \ldots, z_N$: cluster assignments K: number of clusters
1	$K = 1$, $z_i = 1$ for all $i \in 1, \ldots, N$	**1**	$K = 1$, $z_i = 1$ for all $i \in 1, \ldots, N$
2	$E_{\text{new}} = \infty$	**2**	$E_{\text{new}} = \infty$
3	**repeat**	**3**	**repeat**
4	$E_{\text{old}} = E_{\text{new}}$	**4**	$E_{\text{old}} = E_{\text{new}}$
5	**for** $i \in 1, \ldots, N$	**5**	**for** $i \in 1, \ldots, N$
6	**for** $k \in 1, \ldots, K$	**6**	**for** $k \in 1, \ldots, K$
7		**7**	$\dot{\sigma}_k^{-i} = \left(\frac{1}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2} N_k^{-i}\right)^{-1}$
8		**8**	$\dot{\mu}_k^{-i} = \sigma_k^{-i}\left(\frac{\mu_0}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2}\sum_{j:z_j=k, j\neq i} x_j\right)$
9	$d_{i,k} = \frac{1}{2}\left\|x_i - \mu_k\right\|_2^2$	**9**	$d_{i,k} = \frac{1}{2\left(\dot{\sigma}_k^{-i} + \hat{\sigma}^2\right)}\left\|x_i - \dot{\mu}_k^{-i}\right\|_2^2 + \frac{D}{2}\ln\left(\dot{\sigma}_k^{-i} + \hat{\sigma}^2\right)$
10	$d_{i,K+1} = \lambda$	**10**	$d_{i,K+1} = \frac{1}{2\left(\sigma_0^2 + \hat{\sigma}^2\right)}\left\|x_i - \mu_0\right\|_2^2 + \frac{D}{2}\ln\left(\sigma_0^2 + \hat{\sigma}^2\right)$
11	$z_i = \arg\min_{k \in 1, \ldots, K} d_{i,k}$	**11**	$z_i = \arg\min_{k \in 1, \ldots, K+1}\left[d_{i,k} - \ln N_k^{-i}\right]$
12	**if** $z_i = K + 1$	**12**	**if** $z_i = K + 1$
13	$\mu_{K+1} = x_i$	**13**	
14	$K = K + 1$	**14**	$K = K + 1$
15	**for** $k \in 1, \ldots, K$	**15**	
16	$\mu_k = \frac{1}{N_k}\sum_{j:z_j=k} x_j$	**16**	
17	$E_{\text{new}} = \sum_{k=1}^{K}\sum_{i:z_i=k} d_{i,k}$	**17**	$E_{\text{new}} = \sum_{k=1}^{K}\sum_{i:z_i=k} d_{i,k}$ $-K\ln N_0 - \sum_{k=1}^{K}\log\Gamma\left(N_k\right)$
18	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$	**18**	**until** $E_{\text{old}} - E_{\text{new}} < \epsilon$

Let us consider the exponential family DPMM described in Equation (3.13). The mixing parameters and the component parameters are integrated out, so the only random variables left are the indicators $z_1, \ldots, z_N$. Under the MAP framework we suggest iterating through each of the cluster indicators z_i and updating them with their respective MAP values. For each point i, we ignore the normalization terms (the terms independent of k) and compute the negative log of the assignment probabilities from (3.14):

$$q_{i,k} = \psi_0\left(\tau_0 + \sum_{j:z_j=k, j\neq i} g\left(x_j\right), \eta_0 + N_k^{-i}\right) - \psi_0\left(\tau_0 + \sum_{j:z_j=k} g\left(x_j\right), \eta_0 + N_k\right) - \ln N_k^{-i} \tag{3.31}$$

$$q_{i,K+1} = \psi_0\left(\tau_0, \eta_0\right) - \psi_0\left(\tau_0 + g\left(x_i\right), \eta_0 + 1\right) - \ln N_0 \tag{3.32}$$

where by substituting $\tau_k^{-i} = \tau_0 + \sum_{j:z_j=k, j\neq i} g\left(x_j\right)$; $\eta_k^{-i} = \eta_0 + N_k^{-i}$; $\tau_k = \tau_0 + \sum_{j:z_j=k} g\left(x_j\right)$ and $\eta_k = \eta_0 + N_k$, we get the update for existing clusters $q_{i,k} = \psi_0\left(\tau_k^{-i}, \eta_k^{-i}\right) - \psi_0\left(\tau_k, \eta_k\right) - \ln N_k^{-i}$. We have used the same notation as in Section 3.6.1: $g\left(\cdot\right)$ is the sufficient statistics function; $\left(\tau_0, \eta_0\right)$ are prior component hyperparameters and $\psi_0\left(\cdot\right)$ is the log-parition function obtained after integrating out the parameter space. For each observation x_i we compute the above $K+1$-dimensional vector q_i and select the cluster number according to the following:

$$z_i = \underset{k\in\{1,\ldots,K,K+1\}}{\arg\min}\ q_{i,k}$$

The algorithm proceeds to the next observation x_{i+1} by updating the cluster component statistics (posterior hyperparameters) to reflect the new value of the cluster assignment z_i. To check convergence of the algorithm we compute the complete data likelihood:

$$p\left(x, z|N_0\right) = \left(\prod_{i=1}^{N}\prod_{k=1}^{K} p\left(x_i|z_i\right)^{\delta_{z_i,k}}\right) p\left(z_1, \ldots, z_N\right) \tag{3.33}$$

where $\delta_{z_i,k}$ is the Kronecker delta and $p\left(z_1, \ldots z_N\right)$ is the probability of partitions induced by the CRP (Pitman & Yor, 1997) also given in (2.43); computationally it is more convenient to evaluate the negative log of the data likelihood (the NLL), which would lead to equivalent convergence tests. In Appendix A we have included a table describing the choice for sufficient statistics, base measure, log-partition function and hyperparameters depending on the choice of data likelihood. For pedagogical purposes, we have provided a collapsed MAP-DPMM implementation for spherical Gaussian data (Algorithm **3.6**).

It is worth pointing out that unlike MCMC approaches, iterative MAP methods do not increase the negative log of the complete data likelihood at each step and as a result are guaranteed to converge to a fixed point. Convergence is quickly reached. The main disadvantage with this is that the solution at convergence is only guaranteed to be a local minima. Multiple restarts using random permutations of the data can be used to search for improved solutions. However, unlike SVA approaches, with collapsed MAP-DPMM (and with MAP-DPMM) it is possible to learn all model hyperparameters as we discuss in Appendix E.

3.8.2 The MAP-DPMM algorithm

Consider now we start from the CRP construction of the DPMM from Section 3.6.1 which refers explicitly to the component parameters. MAP-DPMM then involves iterating through each of the cluster indicators z_i and also through each of the cluster parameters θ (which in collapsed MAP-DPMM were integrated out), updating them with their respective MAP values at each step. For each point i, we compute for each existing cluster k and for a new cluster $K+1$:

$$q_{i,k} = \psi(\theta_k) + h(x_i) - \langle g(x_i), \theta_k \rangle - \ln N_k^{-i} \tag{3.34}$$

$$q_{i,K+1} = \psi_0(\tau_0, \eta_0) - \psi_0(\tau_0 + g(x_i), \eta_0 + 1) - \ln N_0 \tag{3.35}$$

with again $g(\cdot)$ denoting the sufficient statistics function; $\psi(\cdot)$ being the log partition function and $h(x_i)$ being the base measure. For each observation x_i we obtain a $K+1$-dimensional vector q_i and select the cluster number according to the following:

$$z_i = \underset{k \in \{1,\dots,K,K+1\}}{\arg\min} q_{i,k} \tag{3.36}$$

If $z_i = K+1$ has been chosen, we introduce new set of parameters θ_{K+1}, where we simply choose the values for θ_{K+1} that maximize the posterior of θ given single point x_i. Equivalently, we can choose the values for θ_{K+1} that minimize the negative log posterior of θ given x_i which is easier to compute.

After we sweep through the cluster indicators, we proceed to update the component parameters θ. For $k = 1, \dots, K$:

$$\theta_k = \underset{\theta_k}{\arg\min} \left[\eta_k \psi(\theta_k) + \psi_0(\boldsymbol{\tau}_k, \eta_k) - \langle \theta_k, \boldsymbol{\tau}_k \rangle\right] \tag{3.37}$$

3.8.3 Out-of-sample prediction

The major advantage of the MAP approach compared to SVA is that it will allow us to do *out-of-sample prediction* about unseen observations in a neat and rigorous way. To compute the out-of-sample likelihood for a new observation x_{N+1} we consider two approaches that differ in how the indicator z_{N+1} is treated:

1. *Mixture predictive density.* The unknown indicator z_{N+1} can be integrated out resulting in a mixture density:

$$p(x_{N+1}|N_0, z, x) = \sum_{k=1}^{K+1} p(z_{N+1} = k|N_0, z, X)\, p(x_{N+1}|z_{N+1} = k, \dots) \tag{3.38}$$

   The assignment density $p(z_{N+1} = k|z, N_0, x)$ here is simply computed using the CRP defining rule (2.42), therefore probability for an existing cluster is $\frac{N_k}{N_0+N}$ and $\frac{N_0}{N_0+N}$ is the probability for a new cluster. The second term will be computed differently if we are to use MAP-DPMM or collapsed MAP-DPMM. For the collapsed MAP-DPMM the second term corresponds to the predictive distribution of a point x_{N+1} according to the predictive densities for new and existing cluster:

$$p(x_{N+1}|z_{N+1} = k, \dots) = \begin{cases} p(x_{N+1}|z_{-i}, x, z_{N+1} = k, \tau_k, \eta_k) & \text{for existing } k \\ p(x_{N+1}|\tau, \eta, z_{N+1} = K+1) & \text{for new cluster} \end{cases} \tag{3.39}$$

   For the non-collapsed MAP-DPMM the second term will take different form for existing and new cluster where for existing cluster k we compute the likelihood of point x_{N+1} given it is from the k-th component and for a new cluster we compute the same predictive distribution as in the collapsed MAP-DPMM:

$$p(x_{N+1}|z_{N+1} = k, \dots) = \begin{cases} p(x_{N+1}|\theta_k, z_{N+1} = k) & \text{for existing } k \\ p(x_{N+1}|\tau, \eta, z_{N+1} = K+1) & \text{for new cluster} \end{cases} \tag{3.40}$$

2. *MAP cluster assignment.* We can also use a point estimate for z_{N+1} by picking the minimum negative log posterior of the indicator $p\left(z_{N+1}|x_{N+1}, z, N_0\right)$, equivalently:

$$z_{N+1}^{\text{MAP}} = \underset{k\in\{1,\dots,K,K+1\}}{\arg\min} \left[-\log p\left(x_{N+1}|z_{N+1}=k,\dots\right) - \log p\left(z_{N+1}=k|N_0,z,x\right)\right] \quad (3.41)$$

where $p\left(x_{N+1}|z, X, z_{N+1}=k\right)$ and $p\left(z_{N+1}=k|N_0,z,x\right)$ are computed as in the first approach.

The first (marginalization) approach is used in (Blei & Jordan, 2006) and is more 'robust' as it incorporates the probability of all cluster components while the second (modal) approach can be useful in cases where only a point cluster assignment is needed. Even when using the first approach however, the mixture density is still computed assuming point assignments for the training data $z_1,\dots,z_N$. The resulting approximation error decreases for large sample sizes and for well-separated clusters. This is because ignoring the uncertainty in the training data assignments has a reduced effect on the estimation of the predictive density.

Furthermore, the predictive density obtained using iterative MAP will be comparable to the one obtained using the Gibbs sampler inference only when the sufficient statistics $N_1,\dots N_K$ of the categorical likelihood for the assignment variables estimated from a Gibbs chain are similar to the ones estimated from the modal estimates for $z_1,\dots,z_N$. Empirically, we have observed this often to be the case. We have noticed that the predictive density for highly populated cluster components tend to be well approximated by iterative MAP where the effect of the smaller cluster components diminishes when using only modal estimates for z. Note that the DPMM usually models data with a lot of inconsistent and small, yet spurious components Miller & Harrison (2013); those and any consistent components with small effect are likely to be ignored when using MAP inference as we later show in Section 3.9.2. To summarize, using only modal estimates for the cluster assignments we are likely to infer correctly only the larger components which have a large effect on the model likelihood and which will also affect the estimated predictive density accordingly.

3.8.4 Analysis of iterative MAP for DPMM

Despite being derived from the same model (DPMM), both MAP-DPMM and collapsed MAP-DPMM can produce different clustering results as they imply different treatment for some of the variables in the underlying model. In Gibbs sampling Rao-Blackwellization of some intermediate random variables can improve the mixing of the sampler while converging to an equivalent stationary distribution. However, when deriving MAP algorithms, integrating out random variables from the probabilistic model leads to different clustering procedures. MAP methods do not have the asymptotic guarantees of MCMC techniques, therefore MAP-DPMM and collapsed MAP-DPMM could easily tolerate non-equivalent local solution despite starting from the same model.

For example, let us consider the special case of a DPMM with Gaussian components and follow the derivation of the corresponding MAP-DPMM and collapsed MAP-DPMM algorithms. To derive collapsed MAP-DPMM we integrate over both the mixture weights π and the component parameters $\theta=(\mu,\Sigma)$(which are the means and the covariances for Gaussian data). Therefore, in the new collapsed model each component is described with a Student-t distribution parametrized by some statistics from the data. Recall that in a model where the component parameters (μ,Σ) are not integrated, each component is described instead with a Gaussian distribution, but with random (unknown) parameters. The Student-t distribution has an extra degree of freedom compared to the Gaussian and it typically places higher probability on values that are far from the sample mean (values that are in the tails of the distribution). More specifically, we can see (A.5) in Appendix A that the degrees of freedom $a_k = a_0 + {}^{N_k}/_2$ of each Student-t component depend on the

number of points assigned to that component. As N_k gets larger, the corresponding Student-t component approaches the behaviour of a Gaussian distribution; for components with smaller N_k (with less observations) the Student-t differs more and places higher probability in the tails (see Figure 3.3). The result is that collapsed MAP-DPMM models smaller components with higher uncertainty (components represented with less observations).

By contrast, to derive the non-collapsed MAP-DPMM we only integrate out the mixture weights π from the model. Therefore, the represented components are Gaussian with mean μ_k and covaraince Σ_k. The event of a point belonging to a non-represented component however is still modeled with a Student-t with $a_0 + 1/2$ degrees of freedom. The result is that in MAP-DPMM the event of creating a new component is modeled with a heavy-tailed distribution[5], by comparison to the event of assigning a point to a represented component. This was not the case for the collapsed MAP-DPMM where both events are modeled with Student-t distributions; the consequence of this is that non-collapsed MAP-DPMM will be more likely to create new components than the collapsed one.

If we consider the identical Gibbs sampling scenario, the same argument can be made for each iteration: a single iteration of the collapsed Gibbs-DPMM is less likely to intiate new components compared to a single iteration of the non-collapsed Gibbs-DPMM. However, this behaviour is compensated by the additional sampling step of the component parameters and the sampler converges asymptotically to the same stationary distribution. This is not the case for iterative MAP inference because it is deterministic, the component updates in MAP-DPMM are only modal updates and the local optima reached by collapsed MAP-DPMM and non-collapsed MAP-DPMM can be different.

The robustness gained by integrating over the cluster components comes at the price of changing some of the conditional independence properties of the graphical model. This can make the algorithm less compact and more memory demanding. The main difference would be in cases when we would like to store a trained model on some memory-constrained device. Whereas a representation of the model trained with collapsed MAP-DPMM involves storing all cluster indicators and all the data, in non-collapsed MAP-DPMM we would only need to store the cluster indicators and the component parameters, that is $2N$ variables compare to $N+2K$ which for $N \gg K$ can lead to substantial difference in memory requirements. If we decide to use, for example, the stick-breaking construction of the DPMM where the mixing parameters π are not integrated out, the minimal representation required to summarize the model would be even more compact with $3K$ variables required in the Gaussian data.

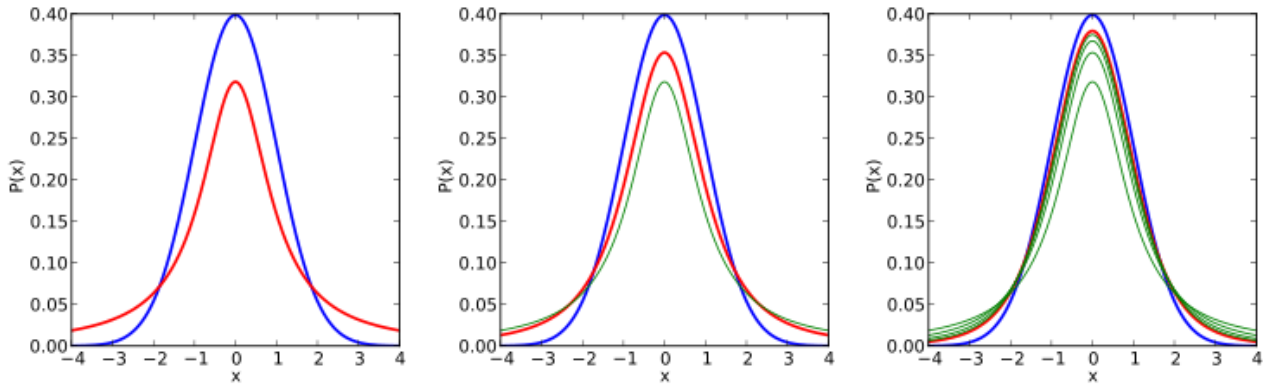
3.9 DPMM experiments

3.9.1 UCI experiment

Next, we compare DP-means, MAP-DPMM, collapsed MAP-DPMM, Gibbs-DPMM and collapsed Gibbs-DPMM on six UCI machine learning repository datasets (Blake & Merz, 1998): *Wine*; *Iris*; *Breast cancer*; *Soybean*; *Pima* and *Vechicle*. We assess the performance of the methods using the same NMI measure as in Section 3.9.2. Class labels in the datasets are treated as cluster numbers[6]. There is either no or a negligibly small number of missing values in each of the data sets. The data types vary between datasets and features: Wine consists of integer and real data; Iris contains real data; Breast cancer consists of integer and categorical

[5] Distributions with heavier tails place higher probability on events far from the main "body" of the distribution than e.g. distributions with exponential tails such as the Gaussian.

[6] We do not assess "Car" and "Balance scale" datasets used in Kulis & Jordan (2011) because they consist of a complete enumeration of 6 and 4 categorical factors respectively, and it is not meaningful to apply an unsupervised clustering algorithm to such a setting.



(a) 1 degree of freedom (b) 2 degrees of freedom (c) 5 degrees of freedom

Figure 3.3: Density of the Student-t distribution with varying degrees of freedom(red line) compared to standard normal distribution(blue line). The green line shows all previous plots with smaller df.

data; Soybean is categorical data; Pima is real data and Vehicle consists of integer data.

To conduct the experiment we have assumed a DPMM with elliptical Gaussian components with unknown means and variances in each dimension. Note that more accurate performance may be reached for MAP-DPMM, collapsed MAP-DPMM and both Gibbs samplers if we carefully pick a different DPMM for each of the datasets that incorporates some knowledge of the type of data. The same initialization conditions are used for all of the algorithms except for DP-means where such initialization is not applicable. For Gibbs sampling we report average NMI score and standard deviation in the brackets, which have been estimated after *burning-in*[7] first 150 iterations. Convergence for the Gibbs samplers is assessed using Raftery and Lewis (Raftery & Lewis, 1992) diagnostics. Mirroring Kulis & Jordan (2011), the threshold parameter λ for DP-means was chosen to give the true number of clusters in each of the datasets.The reported CPU run times for the compared methods were obtained on Matlab R2013a (8.1.0.604) 64-bit (glnxa64), i7-2600 CPU with 3.40GHz processor, ubuntu PC.

Wine data

This dataset is formed of 178 data points each with 13 attributes[8]. The data depicts the outcome of chemical analysis of wines grown in the same region which are from three different cultivars. The chemical analysis determines the values of the 13 attributes of each wine and the wines can be classified into three classes depending on the cultivar. We cluster the data with different algorithms (ignoring the labels for type of cultivar) and use the type of cultivar as a label for the ground truth clustering when evaluating the performance of the methods.

Iris data

This data consists of 150 instances of iris plants each described with 4 attributes: sepal length (in cm); sepal width (in cm); petal length (in cm); petal width (in cm). The iris plants can be classified into three different types of iris plant. We cluster the iris data with different algorithms aiming at correctly distinguishing the three types of plants and we use the true labels to evaluate the performance of the methods.

[7]The practice of throwing away the first few iterations of a MCMC run to remove the most highly-correlated samples.

[8]The attributes in the Wine dataset are: alcohol; malic acid; ash; alcalinity of ash; magnesium; total phenols; flavanoids; nonflacanoid phenols; proanthocyanins; color intensity; hue; OD280/OD315 of diluted wines; proline.

Table 3.3: Clustering performance of DPMM inference techniques measured using NMI between estimated and ground truth labels of the wine dataset from the UCI machine learning repository

Model/Score	NMI	Iterations to convergence	CPU time (seconds)	Number of clusters
Gibbs-DPMM	0.75 (0.04)	1017	415.7	6
Collapsed Gibbs-DPMM	0.74 (0.04)	907	194.8	5
MAP-DPMM	0.56	7	0.48	2
Collapsed MAP-DPMM	0.52	5	0.92	2
DP-means	0.42	19	~	~

Table 3.4: Clustering performance of DPMM inference techniques measured using NMI between estimated and ground truth labels of the iris dataset from the UCI machine learning repository

Model/Score	NMI	Iterations to convergence	CPU time (seconds)	Number of clusters
Gibbs-DPMM	0.77 (0.03)	1017	69.6	4
Collapsed Gibbs-DPMM	0.72 (0.02)	1118	155.3	4
MAP-DPMM	0.73	5	0.75	2
Collapsed MAP-DPMM	0.75	5	0.52	2
DP-means	0.76	8	~	~

Breast cancer data

This dataset consists of data for 286 breast cancer patients with 9 attributes recorded for each participant. The attributes include: age; menopause; tumor size; inv-nodes; node-caps; deg-malig; breast(left or right); breast-quad (which quadrant is the tumor in); irradiate. The patients can be classified into cases with recurrent events and cases with no recurrence. We cluster the patient attributes and aim to correctly classify in an unsupervised way whether a case should belong in the group with recurrence or without.

Soybean data

The original soybean dataset in the UCI repository consists of 307 instances each with 35 attributes of the soybean plant. The plants can be classified into 19 types, where commonly only 15 of those are used for classification and clustering experiments. This is because data from the other 4 types is not well represented in this dataset. In our experiment we follow the same logic and evaluate our methods on the plants from the 15 well-represented types of soybean. There are 266 such instances of soybean plant. The attributes are categorical and vary from features related to the weather and conditions in which the plant grows to features related to the plant's leaves, stems, fruit-pods among others. Data for the plants is again clustering using

Table 3.5: Clustering performance of DPMM inference techniques measured using NMI between estimated and ground truth labels of the breast cancer dataset from the UCI machine learning repository

Model/Score	NMI	Iterations to convergence	CPU time (seconds)	Number of clusters
Gibbs-DPMM	0.61 (0.05)	1499	2760.3	4
Collapsed Gibbs-DPMM	0.73 (0.01)	1023	681.4	2
MAP-DPMM	0.73	5	3.0	2
Collapsed MAP-DPMM	0.75	6	2.1	2
DP-means	0.75	8	~	~

Table 3.6: Clustering performance of DPMM inference techniques measured using NMI between estimated and ground truth labels of the soybean dataset from the UCI machine learning repository

Model/Score	NMI	Iterations to convergence	CPU time (seconds)	Number of clusters
Gibbs-DPMM	0.33 (0.01)	982	2760	4
Collapsed Gibbs-DPMM	0.52 (0.01)	907	681	2
MAP-DPMM	0.36	3	3	2
Collapsed MAP-DPMM	0.38	5	2	2
DP-means	0.36	14	~	~

Table 3.7: Clustering performance of DPMM inference techniques measured using NMI between estimated and ground truth labels of the Pima dataset from the UCI machine learning repository

Model/Score	NMI	Iterations to convergence	CPU time (seconds)	Number of clusters
Gibbs-DPMM	0.06 (0.01)	1248	1289	4
Collapsed Gibbs-DPMM	0.06 (0.01)	907	1097	2
MAP-DPMM	0.05	10	14 seconds	2
Collapsed MAP-DPMM	0.04	6	3 seconds	2
DP-means	0.03	19	~	~

different inference algorithms for DPMM and performance is evaluated using the true labels denoting the soybean type.

Pima data

This dataset consist of 8 attributes of 768 female participants who are aged at least 21 and have Pima Indian heritage. The 8 features that are included for each patient are: Number of times pregnant; plasma glucose concentration; diastolic blood pressure; triceps skin fold thickness; 2-hour serum insulin; body mass index; diabetes pedigree function and age. The participants are classified into two classes depending or whether they are diagnosed with diabetes or not (class 0 and 1). We cluster the patient's data aiming to correctly estimate diabetes diagnosis and then evaluate the different algorithms using the true labels.

Discussion

The study above demonstrates that on four out of six datasets, iterative MAP methods produce comparable clustering results to the clustering produced using exhaustive Gibbs sampling. At the same time, Gibbs sampling requires approximately three orders of magnitude more iterations to converge. On all six datasets the MAP methods perform as well as (often even better than) DP-means algorithms despite the fact that DP-means has been given the true number of clusters a-priori. DP-means performs well on lower-dimensional data sets with a small number of clusters. For higher dimensional data it is more often the case that the different features have different numerical scaling, so the squared Euclidean distance used in DP-means is often inappropriate.

Consider now a more flexible DPMM with unknown mean and complete covariance. The elliptical model does not model any correlation between the attributes of the data implying they are independent[9]. In Table 3.8 we plot the NMI score evaluated using the new clustering produced using collapsed MAP-DPMM inference and matching collapsed Gibbs-DPMM for Gaussian DPMM modeling using complete component

[9]The elliptical Normal-Gamma DPMM is only a special case of the full covariance Normal-Wishart DPMM

Table 3.8: Clustering performance of collapsed MAP-DPMM and collapsed Gibbs-DPMM sampling inference applied to Gaussian DPMM with complete covariances. Performance is evaluated using NMI between estimated and ground truth labels. The algorithms are tested on six UCI datasets. In square brackets is the number of iteration to convergence for the collapsed MAP-DPMM. In round brackets is the standard deviation of the NMI score for different samples of the chain simulated from the collapsed Gibbs-DPMM. The table does not include iterations to convergence for the Gibbs sampler due to the similarity of the figure with the iterations reported from the earlier sampler in Tables 1.3-1.7

Dataset/Model	Collapsed MAP-DPMM	Collapsed Gibbs-DPMM
Wine	0.86 [11]	0.72 (0.06)
Iris	0.78 [5]	0.75 (0.06)
Breast cancer	0.76 [8]	0.73 (0.01)
Soybean	0.40 [9]	0.49 (0.00)
Pima	0.07 [17]	0.07 (0.01)

covariance matrices. In principle this full covariance DPMM does not describe the data significantly better than the simpler elliptical model. Evidence for this is the minimal improvement in the clustering produced by the Gibbs sampler. However, deterministic MAP algorithms derived from this model are less likely to get stuck in poor local optima. The clustering performance of the collapsed MAP-DPMM applied to the more general DPMM is significantly higher, gets close to the performance of the Gibbs sampler and significantly outperforms the DP-means algorithm.

3.9.2 Synthetic CRP parameter estimation

Next, we will examine the performance of collapsed MAP-DPMM, collapsed Gibbs-DPMM, DP-means and VB-DPMM (Section 3.7.1) on synthetically generated CRP-partitioned, non-spherical Gaussian data in terms of estimation error and the computational effort. We generate 100 samples from a two-dimensional DPMM. The partitioning is sampled from a CRP with fixed concentration parameter $N_0 = 3$ and data size $N = 600$. Gaussian component parameters are sampled from a Normal-Wishart (NW) prior with parameters $m_0 = [2, 3]$, $c_0 = 0.5$, $a_0 = 30$, $B_0 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$. This prior ensures a combination of both well-separated and overlapping clusters. We fit the model with collapsed MAP-DPMM, VB-DPMM and collapsed Gibbs-DPMM using the ground truth model hyperparameters (which are the NW parameters (m_0, c_0, a_0, B_0) and the concentration parameter N_0) used to generate the data. Convergence for the Gibbs sampler is tested using the Raftery diagnostic with parameter values $q = 0.025, r = 0.1, s = 0.95$. We use a high convergence acceptance tolerance of $r = 0.1$ to obtain less conservative estimates for the number of iterations required. We use the most likely value from the Gibbs chain after burn-in samples (1/3 of the samples) have been removed.

Clustering estimation accuracy is again measured using NMI score. The parameter λ for DP-means is set using a binary search procedure such that the algorithm gives rise to the correct number of partitions (see Appendix G). This approach again favours DP-means as it is given knowledge of the true number of clusters. For VB-DPMM we set the truncation limit to ten times the number of clusters in the current CRP sample.

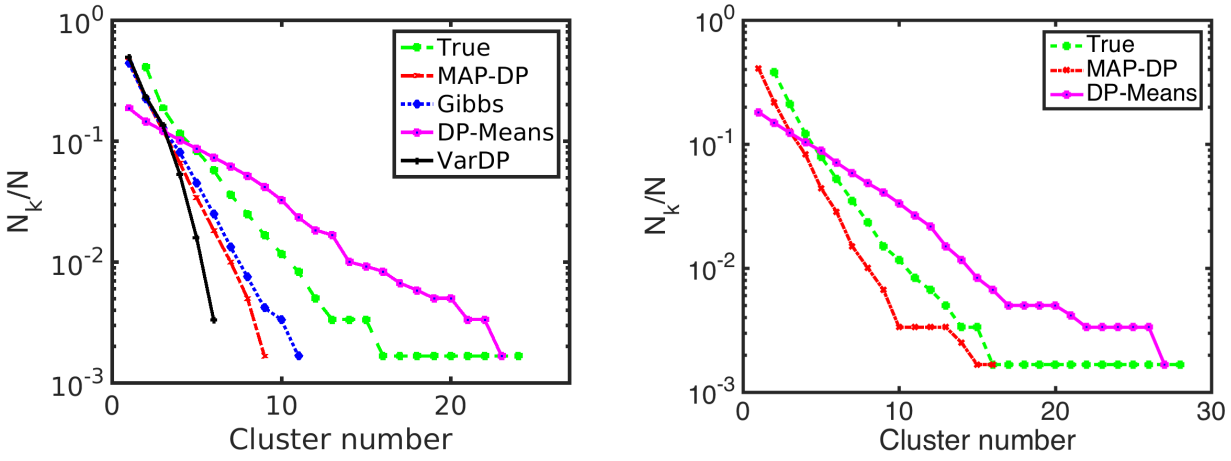
Both the collapsed MAP-DPMM and Gibbs-DPMM achieve similar clustering performance in terms of NMI whilst VB-DPMM and DP-means have lower scores (Table 3.9). Collapsed MAP-DPMM requires the smallest number of iterations to converge with collapsed Gibbs-DPMM requiring, on average, 140 times more iterations and DP-means 1.8 times more. In Figure 3.4(a) the median partitioning is shown in terms of the partitioning N_k/N and the number of clusters. As expected, when using a CRP prior, the sizes of the

Table 3.9: Performance of collapsed Gibbs-DPMM, collapsed MAP-DPMM, DP-means and VB-DPMM inference methods used for clustering synthethic DPMM distributed data. The first row plots average NMI score and standard deviation (in brackets) across the 100 DPMM samples. The second row reports respectively the average number of iterations to convergence and standard deviation (in brackets). Note that the NMI score and iterations to convergence reported for the collapsed Gibbs-DPMM here are measured for the draw of the simulated chain which scores the highest likelihood.

Diagnostic/Model	Collapsed Gibbs-DPMM*	Collapsed MAP-DPMM	DP-means	VB-DPMM
NMI score	0.81 (0.1)	0.82 (0.1)	0.68 (0.1)	0.75 (0.1)
Iterations	1395 (651)	10 (3)	18 (7)	45 (18)

different clusters vary significantly with many small clusters containing only a few observations. Collapsed MAP-DPMM and VB-DPMM fail to identify the smaller clusters whereas the Gibbs sampler is able to do so to a greater extent. This is a form of underfitting where the algorithm captures the mode of the partitioning distribution but fails to put enough mass on the tails (the smaller clusters). The NMI scores do not reflect this effect as the impact of the smaller clusters on the overall measure is minimal. The poorer performance of the DP-means algorithm can be attributed to the non-spherical nature of the data as well as the lack of reinforcement effect that leads to underestimation of the larger clusters and overestimation of the smaller clusters.

To confirm that the performance of DP-means suffers due to the lack of reinforcement effect in its assignment criteria we modify the CRP experiment to sample from spherical clusters (Figure 3.4(b)). CRP-distributed indicators are again sampled 100 times and the collapsed MAP-DPMM algorithm attains NMI scores of 0.88 (0.1) and DP-means scores NMI 0.73 (0.1). As the clusters are spherical, the lower performance of the DP-means algorithms is solely explained by the lack of reinforcement effect.



(a) Non-spherical (b) Spherical

Figure 3.4: CRP mixture experiment; distribution of cluster sizes, actual and estimated using different methods. Cluster number ordered by decreasing size (horizontal axis) vs $\frac{N_k}{N}$ (vertical axis).

3.10 Example applications of MAP-DPMM algorithms

3.10.1 Sub-typing of parkinsonism and Parkinson's disease

Parkinsonism is the clinical syndrome defined by the combination of bradykinesia (slowness of movement) with tremor, rigidity or postural instability. This clinical syndrome is most commonly caused by *Parkinson's disease* (PD), although can be caused by drugs or other conditions such as multi-system atrophy. Because of the common clinical features shared by these other causes of parkinsonism, the clinical diagnosis of PD in vivo is only 90% accurate when compared to post-mortem studies. This diagnostic difficulty is compounded by the fact that PD itself is a heterogeneous condition with a wide variety of clinical phenotypes, likely driven by different disease processes. These include wide variations in both the *motor* (movement, such as tremor and gait) and *non-motor* symptoms (such as cognition and sleep disorders). While the motor symptoms are more specific to parkinsonism, many of the non-motor symptoms associated with PD are common in older patients which makes clustering these symptoms more complex. Despite significant advances, the aetiology (underlying cause) and pathogenesis (how the disease develops) of this disease remain poorly understood, and no disease modifying treatment has yet been found.

The diagnosis of PD is therefore likely to be given to some patients with other causes of their symptoms. Also, even with the correct diagnosis of PD, they are likely to be affected by different disease mechanisms which may vary in their response to treatments, thus reducing the power of clinical trials. Despite numerous attempts to classify PD into sub-types using empirical or data-driven approaches (using mainly K-means cluster analysis), there is no widely accepted consensus on classification.

One approach to identifying PD and its subtypes would be through appropriate clustering techniques applied to comprehensive datasets representing many of the physiological, genetic and behavioural features of patients with parkinsonism. We expect that a clustering technique should be able to identify PD subtypes as distinct from other conditions. In that context, using methods like K-means and finite mixture models would severely limit our analysis as we would need to fix a-priori the number of sub-types K for which we are looking. Estimating that K is still an open question in PD research. Potentially, the number of sub-types is not even fixed, instead, with increasing amounts of clinical data on patients being collected, we might expect a growing number of variants of the disease to be observed. A natural probabilistic model which incorporates that assumption is the DP mixture model. Here we make use of collapsed MAP-DPMM as a computationally convenient alternative to fitting the DP mixture.

We have analyzed the data for 527 patients from the *PD data and organizing center* (PD-DOC) clinical reference database, which was developed to facilitate the planning, study design, and statistical analysis of PD-related data (Kurlan & Murphy, 2007). The subjects consisted of patients referred with suspected parkinsonism thought to be caused by PD. Each patient was rated by a specialist on a percentage probability of having PD, with 90-100% considered as probable PD (this variable was not included in the analysis). This data was collected by several independent clinical centers in the US, and organized by the University of Rochester, NY. Ethical approval was obtained by the independent ethical review boards of each of the participating centers. From that database, we use the PostCEPT data.

For each patient with parkinsonism there is a comprehensive set of features collected through various questionnaires and clinical tests, in total 215 features per patient. The features are of different types such as yes/no questions, finite ordinal numerical rating scales, and others, each of which can be appropriately modeled by e.g. Bernoulli (yes/no), binomial (ordinal), categorical (nominal) and Poisson (count) random variables (see Appendix A). For simplicity and interpretability, we assume the different features are independent and use the elliptical model defined in Section 3.8.

Table 3.10: Significant features of parkinsonism from the PostCEPT/PD-DOC clinical reference data across clusters (groups) obtained using collapsed MAP-DPMM with appropriate distributional models for each feature. Each entry in the table is the probability of PostCEPT parkinsonism patient answering "yes" in each cluster (group).

	Group 1	Group 2	Group 3	Group 4
Resting tremor (present and typical)	0.81	0.91	0.42	0.78
Resting tremor (absent)	0.14	0.06	0.42	0.11
Symptoms in the past week	0.58	0.94	1.00	0.67

Table 3.11: Significant features of parkinsonism from the PostCEPT/PD-DOC clinical reference data across clusters obtained using collapsed MAP-DPMM with appropriate distributional models for each feature. Each entry in the table is the mean score of the ordinal data in each row. Lower numbers denote condition closer to healthy. Note that the Hoehn and Yahr stage is re-mapped from $\{0, 1.0, 1.5, 2, 2.5, 3, 4, 5\}$ to $\{0, 1, 2, 3, 4, 5, 6, 7\}$ respectively.

Mean score	Scale	Group1	Group 2	Group 3	Group 4
Facial expression	0-4	1.42	1.47	0.42	2.33
Tremor at rest (face, lips and chin)	0-4	0.05	0.32	0.23	1.00
Rigidity (right upper extremity)	0-4	0.90	1.30	0.38	2.11
Rigidity (left upper extremity)	0-4	0.62	1.33	0.19	2.00
Rigidity (right lower extremity)	0-4	0.46	0.97	0.04	2.56
Rigidity (left lower extremity)	0-4	0.38	1.06	0.04	2.67
Finger taps (left hand)	0-4	0.65	1.41	0.50	2.33
PD state during exam	1-4	2.65	3.85	4.00	3.00
Modified Hoehn and Yahr stage	0-7	2.46	3.19	1.62	6.33

A common problem that arises in health informatics is missing data. When using K-means this problem is usually separately addressed prior to clustering by some type of *imputation* method. However, in the MAP-DPMM framework, we can simultaneously address the problems of clustering and missing data. In the probabilistic treatment the missing values are considered as an additional set of random variables and MAP-DPMM proceeds by updating them at every iteration. As a result, the missing values and cluster assignments will depend upon each other so that they are consistent with the observed feature data and each other.

We initialized MAP-DP with 10 randomized permutations of the data and iterated to convergence on each randomized restart. The results (Tables 3.10 and 3.11) suggest that the PostCEPT data is clustered into 5 groups with 50%, 43%, 5%, 1.6% and 0.4% of the data in each cluster. We then performed a Student's t-test at $\alpha = 0.01$ significance level to identify features that differ significantly between clusters. As with most hypothesis tests, we should always be cautious when drawing conclusions, particularly considering that not all of the mathematical assumptions underlying the hypothesis test have necessarily been met. Nevertheless, this analysis suggest that there are 61 features that differ significantly between the two largest clusters. Note that if, for example, none of the features were significantly different between clusters, this would call into question the extent to which the clustering is meaningful at all. We assume that the features differing the most among clusters are the same features that lead the patient data to cluster. By contrast, features that have indistinguishable distributions across the different groups should not have significant influence on the clustering.

We applied the significance test to each pair of clusters excluding the smallest one as it consists of only 2 patients. Exploring the full set of multilevel correlations occurring between 215 features among 4

groups would be a challenging task that would change the focus of this work. We therefore concentrate only on the pairwise-significant features between Groups 1-4, since the hypothesis test has higher power when comparing larger groups of data. The clustering results suggest many other features not reported here that differ significantly between the different pairs of clusters that could be further explored. A full list of the significantly different features, corresponding p-values and effect size in Appendix J. Individual analysis on Group 5 shows that it consists of 2 patients with advanced parkinsonism who are unlikely to have PD itself (both were thought to have less than 50% probability of having PD).

Due to the nature of the study and the fact that very little is yet known about the sub-typing of PD, direct numerical validation of the results is not feasible. The purpose of the study is to learn in a completely unsupervised way, an interpretable clustering on this comprehensive set of patient data, and then interpret the resulting clustering by reference to other sub-typing studies.

Our analysis successfully clustered almost all the patients thought to have PD into the 2 largest groups. Only 4 out of 490 patients (which were thought to have Lewy-body dementia, multi-system atrophy and essential tremor) were included in these 2 groups, each of which had phenotypes very similar to PD. Because the unselected population of parkinsonism included a number of patients with phenotypes very different to PD, it may be that the analysis was therefore unable to distinguish the subtle differences in these cases. The fact that a few cases were not included in these group could be due to: an extreme phenotype of the condition; variability in how subjects filled in the self-rated questionnaires (either comparatively under or over stating symptoms); or that these patients were misclassified by the clinician. The inclusion of patients thought not to have PD in these two groups could also be explained by the above reasons.

Comparing the two groups of PD patients (Groups 1 & 2), group 1 appears to have less severe symptoms across most motor and non-motor measures. Group 2 is consistent with a more aggressive or rapidly progressive form of PD, with a lower ratio of tremor to rigidity symptoms. van Rooden *et al.* (2010) combined the conclusions of some of the most prominent, large-scale studies. Of these studies, 5 distinguished *rigidity-dominant* and *tremor-dominant* profiles (Reijnders *et al.*, 2009; Lewis *et al.*, 2005; Liu *et al.*, 2011; Gasparoli *et al.*, 2002). Our analysis, identifies a two subtype solution most consistent with a less severe tremor dominant group and more severe non-tremor dominant group most consistent with (Gasparoli *et al.*, 2002).

These results demonstrate that even with small datasets that are common in studies on parkinsonism and PD sub-typing, MAP-DPMM is a useful exploratory tool for obtaining insights into the structure of the data and to formulate useful hypothesis for further research.

Although the clinical heterogeneity of PD is well recognized across studies (Hoehn *et al.*, 1998), comparison of clinical sub-types is a challenging task. Studies often concentrate on a limited range of more specific clinical features. For instance, some studies concentrate only on cognitive features or on motor-disorder symptoms (Yang *et al.*, 2014). In addition, typically the cluster analysis is performed with the K-means algorithm and fixing K a-priori might seriously distort the analysis.

It is important to note that the clinical data itself in PD (and other neurodegenerative diseases) has inherent inconsistencies between individual cases which make sub-typing by these methods difficult: the clinical diagnosis of PD is only 90% accurate; medication causes inconsistent variations in the symptoms; clinical assessments (both self rated and clinician administered) are subjective; delayed diagnosis and the (variable) slow progression of the disease makes disease duration inconsistent. Therefore, any kind of partitioning of the data has inherent limitations in how it can be interpreted with respect to the known PD disease process. It may therefore be more appropriate estimate a DP mixture density from the data instead of focusing on the modal point estimates for each cluster.

Our analysis presented here has the additional layer of complexity due to the inclusion of patients with parkinsonism without a clinical diagnosis of PD. This makes differentiating further subtypes of PD more difficult as these are likely to be far more subtle than the differences between the different causes of parkinsonism.

3.10.2 Application of MAP-DPMM to semiparametric mixed effect models

Hierarchical modeling is commonly used in the analysis of *longitudinal health data*[10]. A particular model that is widely used in practice is the *linear mixed effects model*:

$$\begin{aligned} y_i &= X_i\beta_i + \epsilon_{i,j} \\ \beta_i &\sim P \end{aligned} \tag{3.42}$$

where y_i is the observation vector for individual $i \in \{1,\ldots,N\}$, $\epsilon_{i,j} \sim \mathcal{N}\left(0, \tau_\sigma^{-1}\right)$ is the subject-specific observation noise with τ_σ the within-subject precision and P the distribution of the *random effects* β_i (Dunson, 2010). X_i are the inputs for the random effects β_i and the fixed effect regression parameters are equal to the mean of the distribution P. The distribution P is commonly specified to be Gaussian for analytical tractability and computational simplicity. However, the assumption of normality is seldom justified and the assumptions of symmetry and unimodality are often found to be inappropriate (Dunson, 2010).

Semiparametric mixed effects models have been proposed to relax the normality assumption by placing a DPMM prior on P (Kleinman & Ibrahim, 1998). However, inference for such models is usually performed using MCMC requiring large computational resources and careful tuning of algorithmic parameters. This makes MCMC approaches particularly difficult to implement on large data sets. The increasing availability of large longitudinal datasets warrants the investigation of computationally efficient inference approaches such as MAP-DPMM.

We construct the semiparametric mixed effects model, first by placing a DPMM prior on β_i in (3.42). As we are interested explicitly in the component parameters we do not collapse them out and use the non-collapsed MAP-DPMM. We substitute the random effects β_i for the individual data points x_i in the MAP-DPMM from Section 3.8.2. Then further steps are added to update the random effects β_i and within-subject precision τ_σ. The conditional $p\left(\beta_i|\tau_\sigma, z=k, \mu_k, R_k\right)$ for the random effects β_i is:

$$\mathcal{N}\left(\beta_i \left| \left(\tau_\sigma X_i^T X_i + R_k\right)^{-1}\left(\tau_\sigma X_i^T y_i + R_k\mu_k\right), \left(\tau_\sigma X_i^T X_i + R_k\right)^{-1}\right.\right) \tag{3.43}$$

where the conditioning is on the assigned cluster k with mean μ_k and precision R_k. We place a conjugate Gamma prior on the within-subject precision $\tau_\sigma \sim \text{Gamma}\left(a_{\sigma^2}, b_{\sigma^2}\right)$ allowing for the calculation of the conditional posterior:

$$p\left(\tau_\sigma|B, a_{\sigma^2}, b_{\sigma^2}\right) = \text{Gamma}\left(\tau_\sigma \left| a_{\sigma^2} + \frac{N}{2}, b_{\sigma^2} + \frac{1}{2}\sum_{i=1}^{N}\left(y_i - X_i\beta_i\right)^T\left(y_i - X_i\beta_i\right)\right.\right) \tag{3.44}$$

where B is the collection of all random effects $\left(\beta_i\right)_{i=1}^N$. The modes of both conditionals needed for hybrid MAP-DP are easily calculated in addition to the negative log likelihood necessary to check for convergence.

[10]Longitudinal data tracks the same measurement of an individual at different points in time.

Table 3.12: Cross-validated, average held-out likelihood for two models.

Cognitive measures	Depression + ADL
0.364	3.834

English Longitudinal Survey of Ageing

We apply the semiparametric mixed effects model above to the *English longitudinal survey of aging* (ELSA), a large longitudinal survey of older adults aged over 50 in the United Kingdom. ELSA is a large, multi-purpose health study which follows individuals aged 50 years or older (Netuveli *et al.*, 2006). Health-related factors collected include clinical, physical, financial and general well-being. Of primary interest is the effect of the different factors on *quality of life* (QoL) measured using a compound measure of several health and socio-economical indicators. The ELSA survey has been conducted in five waves spanning ten years. In this preliminary study we look at the response of 6,805 individuals across all 5 waves.

We wish to check the hypothesis that measures of cognition such as memory and executive mental function, as estimated by *verbal fluency* performance, are useful predictors of QoL and whether they are more informative than standard measures of *depression* and *activities of daily living* (ADL)[11] that have been found to be statistically significant predictors of QoL (Netuveli *et al.*, 2006). We propose to answer these two questions via selection of two models with different sets of *covariates* (also known as predictor or explanatory variables in regression models). The first model includes depression and ADL as inputs whereas the second model includes measures of cognitive ability, specifically *prospective memory*[12] and verbal fluency. The models are assessed using 5-fold cross-validation and computing the average held-out likelihood, (3.38) in Section 3.8.3.

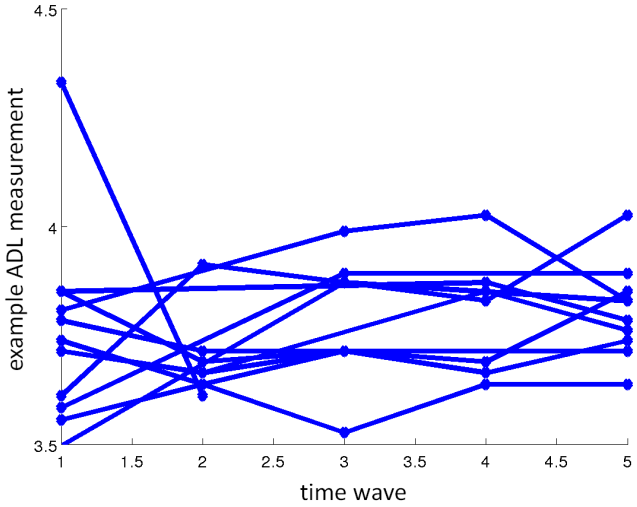


Figure 3.5: Identified cluster for the ELSA longitudinal data.

The model that includes ADL and depression as covariates achieves a significantly lower average held-out likelihood than the competing model containing cognitive measures suggesting that ADL and depression are more informative predictors of QoL than the cognitive measures we considered (Table 3.12).

[11] ADL measures assess an individual's ability to perform basic tasks of everyday life, such as washing and dressing. Measures of depression include various symptoms related to the severity of the desease.

[12] By a "prospective memory" measure we mean a measure of an individual's ability to remember to perform previously planned actions.

The average elapsed time for fitting the model (using iterative MAP inference) that uses ADL and depression as covariates is 11.05 seconds versus 16.29 seconds average elapsed time for fitting the model(again with iterative MAP) with cognitive measures as covariates. The reported run times for MAP and MCMC inference were obtained on Matlab R2013a (8.1.0.604) 64-bit (glnxa64), i7-2600 CPU with 3.40GHz processor, ubuntu PC. For comparison we performed inference using a truncated Gibbs sampler in a DP random effects model: approximately 100,000 iterations were needed to ensure convergence and this is using less than half of the data (3,000 individuals). The resulting time to convergence is in excess of five hours making inference on larger data sets impractical. The rapid inference obtained using extended MAP-DPMM enables a wide array of diagnostic and validation methods to be exploited, which suggests the approach can be scaled up to very large datasets.

3.11 Discussion

In this chapter we studied the properties and the potential benefits of simple deterministic inference algorithms applied to finite and infinite mixture models. Motivated by "folklore" observations in the community and recent work from (Bishop, 2006; Kulis & Jordan, 2011; Jiang *et al.*, 2013; Broderick *et al.*, 2013a) we view ubiquitous clustering techniques such as K-means as simplified methods for inference (SVA inference) in probabilistic models. By formalizing the two separate approaches to SVA reasoning, we fill some existing gaps in the literature and thereby propose novel SVA clustering algorithms which handle unequally distributed data more adequately than algorithms such as K-means (Section 3.2.2 and Section 3.7.2). The chapter also studies how the well-known concept of Rao-Blackwellization can be used to obtain new deterministic inference (in particular clustering) procedures. For example, we derive a novel collapsed K-means algorithm which often converges faster than the original non-collapsed K-means.

In contrast to the SVA inference algorithms, we propose and study another framework for deterministic inference: the iterative MAP method. SVA methods can be seen as a simplified version of iterative MAP inference which although deterministic, preserves a lot more of the useful properties of the underlying probabilistic model (in this chapter we focus particularly on mixture models). We also examine how the process of Rao-Blackwellization can be used to derive different deterministic MAP algorithms.

We thoroughly evaluate the benefits of deterministic MAP inference derived from finite and infinite (DP) mixture models compared to related SVA inference, Gibbs sampling inference and variational Bayes inference. Iterative MAP methods such as MAP-GMM and MAP-DPMM are, practically, as fast as K-means or DP-means, but unlike their SVA counterparts they inherit most of the flexibility and the interpretability of the underlying probabilistic mixture model; MAP-GMM and MAP-DPMM enable model selection and can be used to cluster non-spherical data. Comparing for example MAP-DPMM and Gibbs-DPMM, we see that MAP methods can often obtain as good clustering solution as exhaustive Gibbs, while requiring a few orders less computational effort. By contrast to VB inference methods, iterative MAP does not require additional factorization assumptions about the joint distribution of the model parameters and latent variables, it easier to derive and converges faster.

Probabilistic models such as the DPMM have considerable potential, but are often restricted to applications in which computational resources and time for inference is plentiful. Little effort has been made to develop simple, fast and principled deterministic inference methods that can extend significantly the practical value of probabilistic models(in particular mixture models). We believe methods such as MAP-DPMM are a step forward in this direction and can have many practical applications.

Chapter 4

Deterministic inference and analysis of HDP mixtures

4.1 Introduction

In many real world problems data is often not collected as a single batch and instead it arrives segregated in groups. In natural language processing applications, text data usually is collected from many different documents and we are interested in learning common topics across all documents (Brett, 2012); in biomedecine and bioinformatics data often comes from different studies and experiments where we are interested in learning shared structure across all experiments (Müller *et al.*, 2004; Wang & Wang, 2013); in the problem of subtyping parkinsonisms discussed above (Section 3.10.1) patients are often tested in different hospitals and we are interested in efficiently understanding patients from all hospitals. For such problems it can often be beneficial to incorporate the various groupings of the data into a model hierarchy, rather then mix up all datasets and model data as a single batch.

In this chapter, we review existing work that tries to solve this problem to motivate the derivation of the *hierarchical Dirichlet process* (HDP)(Teh *et al.*, 2006). We discuss different constructions for the HDP as presented in (Teh *et al.*, 2006) and some other ways to do inference in HDP mixtures. Mostly, this is done using different Gibbs samplers, which we review in detail in Section 4.3.3. Such methods restrict the use of the HDP in practice and little work has been done towards developing scalable deterministic methods for inference in HDP models. We extend the earlier SVA and iterative MAP algorithms for inference in HDP mixture models. The potential of our proposed deterministic *MAP-HDP* algorithm is demonstrated empirically on synthetic data in Section 4.5 where it is compared to the SVA approach. The section is concluded with some discussion and review of other approaches to hierarchical clustering and some recent advances to more complex hierarchical modeling.

4.2 Motivation

A lot of progress has been made in using the DPMM to model inherent partitions in a sequence of exchanageable random variables. However, in recent years there is growing interest in extending the DP to accommodate dependent collections of exchangeable random variables Salakhutdinov *et al.* (2013). The problem we wish to solve is this: if we have $j = 1, \ldots, J$ different datasets with exchangeable data but dependence between different datasets, how do we model the partitioning of the data? Let us start by assuming that

each dataset is associated with a separate probability measure G_j drawn from a DP and we denote that DP as $\mathrm{DP}(N_{0,j}, G_{0,j})$. The problem that arises is how to link the different group-specific DPs. A lot of work focuses on describing a relationship between the different DP parameters $N_{0,j}$ and $G_{0,j}$ Cifarelli & Regazzini (1978); Müller *et al.* (2004); Carota & Parmigiani (2002); Muliere & Petrone (1993). For example, (Cifarelli & Regazzini, 1978) and (Muliere & Petrone, 1993) link group-specific measures on the level of the hyper-parameters $N_{0,j}$. This means that the random measures G_j are all drawn from a DP but with different concentration parameters $N_{0,1}, \ldots, N_{0,J}$. This strategy is strictly limited to learning structure that can be reflected through a simple change in the hyperparameter space and does not affect the expectation of each G_j.

A more natural proposal would be to assume that the set specific G_j are all drawn from DPs with the same base measure having $G_{01} \equiv \cdots \equiv G_{0J} \equiv G_0(\theta_0)$ with $G_0(\theta_0)$ being a parametric distribution with random parameter θ_0. However, any arbitrary choice for distribution $G_0(\theta_0)$ will not solve the problem stated above. For example if G_0 is continuous, atoms of the different G_j's will be different with probability 1. If G_0 is a discrete parametric distribution, atoms of the different G_j can be shared, but such an assumption is overly restrictive as in this case the G_j's cannot be used to define infinite mixture models.

4.3 Hierarchical Dirichlet process

In order to force G_0 to be discrete, without fixing its atoms and restricting its support Teh *et al.* suggested assuming that G_0 is itself drawn from a DP, $G_0 \sim \mathrm{DP}(M_0, H)$ with concentration parameter M_0. Although the proposed structure restricts the draws from G_0 to be discrete, G_0 can be nonparametric and its base measure H could itself be a mixture of both discrete and continuous densities. The discreteness of G_0 makes stick-breaking draws easy to construct, and so allows for the atoms of G_0 to be shared among the different data sets. The HDP can be also seen as a specific example of a dependency model for multiple DPs, one specifically aimed at solving the problem of sharing clusters among related groups of data. The HDP can be written in the form:

$$\begin{aligned} G_0 &\sim \mathrm{DP}(M_0, H) \\ G_j &\sim \mathrm{DP}(N_0, G_0) \end{aligned} \tag{4.1}$$

4.3.1 Stick-breaking construction for the HDP

From (4.1), it follows that G_0 is DP distributed, hence using the stick-breaking construction from Section 2.5.2 (Chapter 2), we can express G_0 as an infinite mixture:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}(\cdot) \tag{4.2}$$

with $\beta \sim \mathrm{Stick}(M_0)$ and $\theta_1, \ldots, \theta_\infty$ independent draws from the base measure H. As the group specific G_j all share discrete base measure G_0, they will all have the same atoms as G_0 and we can write:

$$G_j = \sum_{k=1}^{\infty} \pi_{j,k} \delta_{\theta_k}(\cdot) \tag{4.3}$$

where the weights $\pi_{j,1}, \ldots \pi_{j,\infty}$ are independent given β.

To express the stick-breaking process defining the local mixing weights π we go back to the formal definition of the DP (Section 2.5.1). Consider the first K atoms $\theta_1, \ldots, \theta_K$ and define a partition of the base

measure space Θ into K singleton sets $A_1, \ldots, A_K$ with $A_k = \{\theta_k\}$ for $k \in \{1, \ldots, K\}$. Define one more set to cover the rest of the region Θ, $A_{K+1} = \Theta \backslash A_1 \backslash \ldots \backslash A_K$. From the definition of the DP the vector $(G_j(A_1), \ldots, G_j(A_K), G_j(A_{K+1}))$ is distributed as a finite Dirichlet distribution:

$$(G_j(A_1), \ldots, G_j(A_K), G_j(A_{K+1})) \sim \text{Dirichlet}(N_0 G_0(A_1), \ldots, N_0 G_0(A_K), N_0 G_0(A_{K+1})) \tag{4.4}$$

As A_k are singleton sets, $G_j(A_k)$ directly specifies the probability mass on atom θ_k, hence $G_j(A_k) = \pi_{jk}$. Similarly $G_j(A_{K+1})$ determines the probability of an atom being drawn from set A_{K+1} and we can write $G_j(A_{K+1}) = \sum_{l=K+1}^{\infty} \pi_{jl}$ (from the agglomerative property of the Dirichlet distribution).

If we follow the same argument for G_0, we can also write $G_0(A_k) = \beta_k$ for $k = 1, \ldots, K$ and $G_0(A_{K+1}) = \sum_{l=1}^{\infty} \beta_l$. We can now substitute G_j and G_0 into the expression in (4.4) and write:

$$\left(\pi_{j,1}, \ldots, \pi_{j,K}, \sum_{l=K+1}^{\infty} \pi_{j,l}\right) \sim \text{Dirichlet}\left(N_0\beta_1, \ldots, N_0\beta_K, N_0\left(\sum_{l=K+1}^{\infty} \beta_l\right)\right) \tag{4.5}$$

The local weights are Dirichlet distributed around the global weights β with N_0 controlling how different each G_j is from G_0. Teh *et al.* (2006) also showed that (4.5) asymptotically converges to the following stick-breaking construction:

$$\pi_{j,k} = \nu_{j,k} \prod_{l=1}^{K-1} (1 - \beta_{j,l}), \quad \text{with } \nu_{j,k} \sim \text{Beta}\left(N_0\beta_k, N_0\left(1 - \sum_{l=1}^{K} \beta_k\right)\right) \tag{4.6}$$

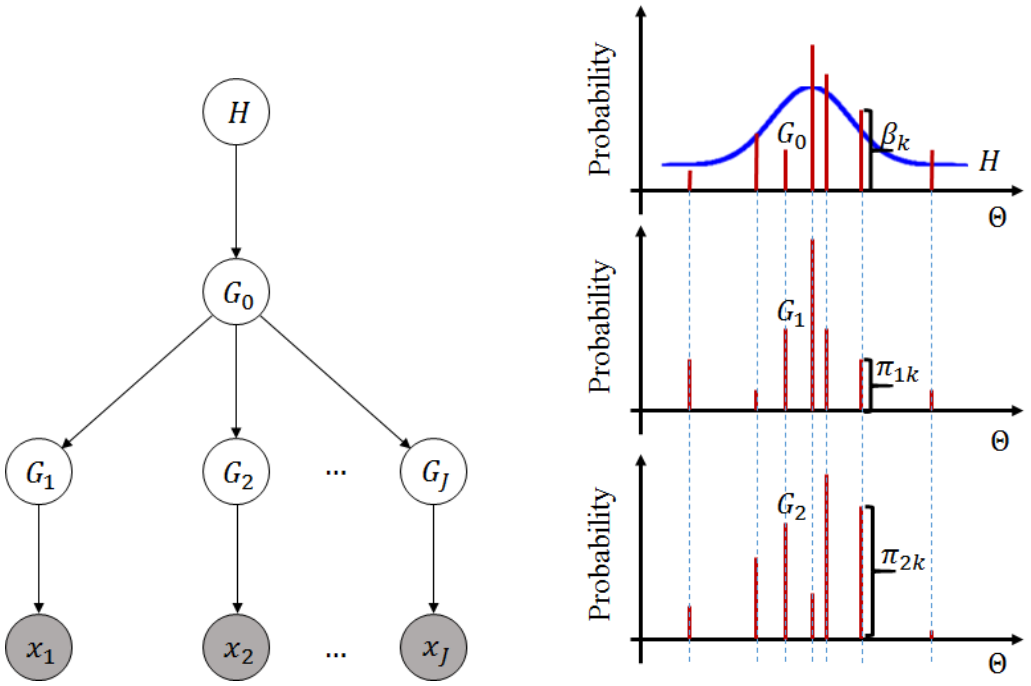
The stick-breaking construction of the HDP can be useful for deriving efficient inference algorithms that do not require integration over the infinite mixing measures (both local and global). In Figure 4.1 we plot an example draw from an HDP with Gaussian base measure H.

4.3.2 Chinese restaurant franchise

Instead of working directly with the HDP, it can often be easier to use its marginal process called the *Chinese restaurant franchise* (CRF) process. In the same way a CRP can be seen as the distribution of the partition induced by a DP after integrating out the random measure, the partition induced by a HDP after integrating out G_0 and $G_1, \ldots, G_J$, is a CRF.

Let us consider the following metaphor as an intuitive description of the CRF stochastic process. Assume that we have a franchise of J restaurants with a shared menu across the branches. At each table in all of the restaurants exactly one dish is served and this dish is chosen by the first customer who is seated there. The served dishes may vary across tables and restaurants but are all from the same menu. In restaurant j, customer i will be seated at table c with probability proportional to the number of customers already seated there, $N_{j,c}$, and with probability proportional to N_0 the customer will be seated at a new table (mirroring a simple CRP). The customer is served the dish θ_k associated with his table unless he is the first one to sit there in which case he chooses a dish. The dish is chosen according to the CRP rule (the top level CRP), where an already served dish k is chosen with probability proportional to the number of times it has been served across all restaurants M_k; a new dish from the menu is chosen with probability proportional to M_0.

Let us now introduce indicator variables denoting where customers sit and what dish they are served: $z_{ji}^{local} = c$ denotes that customer i in restaurant j is sitting at table c; $z_{jc}^{global} = k$ denotes that table c in restaurant j serves dish θ_k. We use N_{jc} to count the number of people seated at table c in restaurant j and M_k counts the number of tables c that serve dish θ_k. We can then express the CRF with the following



(a) Graphical model (b) Stick breaking construction

Figure 4.1: On the left is the graphical model of the hierarchical Dirichlet process (HDP) with x_j denoting vectors of data in set f. On the right is a plot of the global DP G_0 and set specific DPs G_1 and G_2.

conditionals:

$$p\left(z_{j,i}^{local} = c\right) = \begin{cases} \frac{N_{j,c}}{N_0+N_{j,\cdot}} & \text{for occupied table} \\ \frac{N_0}{N_0+N_{j,\cdot}} & \text{for a new table} \end{cases} \tag{4.7}$$

with $c \in \{1, \ldots, C_j\}$ being one of the represented C_j local clusters in set j, $N_{j\cdot} = \sum_{c=1}^{C_j} N_j$ and

$$p\left(z_{j,c}^{global} = k\right) = \begin{cases} \frac{M_k}{M_0+\sum_{j=1}^{K} M_j} & \text{for dish already served} \\ \frac{M_0}{M_0+\sum_{j=1}^{K} M_j} & \text{for a new dish from the menu} \end{cases} \tag{4.8}$$

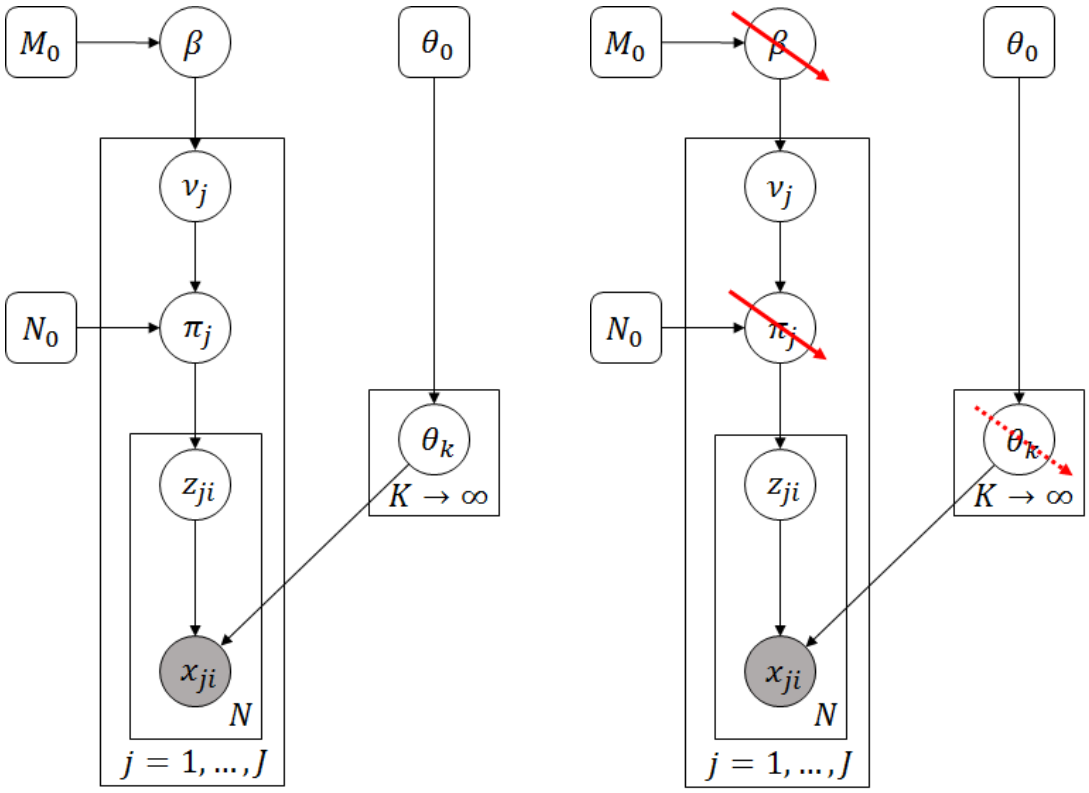
The table indicators imply a local partitioning of the data in each data set and reflect the effect of the local CRPs; the dish indicators imply the shared global partitioning across all sets and reflect the global CRP.

The combination of these nested sets of indicators complicates the bookkeeping during Gibbs inference and Teh *et al.* (2006) has proposed using modified representations of the HDP to simplify this. For example, the *direct assignment* representation bypasses the nested connection of point i from set j to global dish, $z_{ji}^{local} = c \rightarrow z_{jc}^{global} = k$, using direct assignments $z_{ij} = k$ indicating the dish assignment for a customer. Further details can also be found in (Teh *et al.*, 2006) including an *augmented representation* of the HDP.

4.3.3 Gibbs sampling for HDP mixture models

Teh *et al.* (2006) introduced three main Gibbs samplers for inference in the HDP mixture model which make use of different constructions and representations of the HDP prior. All of the three rely on integrating over the set specific measures $G_1, \ldots, G_J$ and work directly on the indicators. *A slice sampling* motivated sampler that works on the full (non-collapsed) HDP space has been introduced in Van Gael *et al.* (2008), but it is

formulated for the application of the HDP to time series (the HDP-HMM) discussed later.



(a) HDP mixture

(b) CRF mixture

Figure 4.2: On the left is the graphical model of the HDP mixture model with explicitly written mixing parameters and corresponding indicators. The random variable β has K dimensions where each of the vectors π_j is associated with a latent variable v_j with the same dimensions specifying component assignments for each element of π_j. π_j consists of the mixing parameter partitioning set j. On the right we show with red arrows the random variables that are integrated out in the CRF construction. On the right is a plot of the global DP G_0 and set specific DPs G_1 and G_2.

CRF-based sampler

The first Gibbs sampler introduced in Teh *et al.* (2006) is referred to as *posterior sampling in the CRF*. As with the CRP-based samplers for the DPMM, in this construction we integrate over the infinite mixing parameters of both the global DP and the local DPs and model the indicator variables explicitly (see Figure 4.2). The integration causes a coupling between the indicator variables z^{local} and z^{global}. For deeper hierarchical models, such coupling can lead to slower mixing compared to a blocked sampling approach which keeps the conditional independence properties of the starting probabilistic graphical model. This is not usually the case for lower level hierarchical models like the DPMM. We have observed that fully collapsing deeper level latent variable models can introduce complex nesting. As a result the corresponding sampler often gets stuck in a region with high probability which is weakly connected to the other high probability regions of the state space. In practice this leads to the sampler becoming dependent (unless we can run it forever) upon the order in which it processes the data. For example, in the case of sequential data problems, collapsed samplers are known to mix quite poorly. Block procedures are more likely to escape poor mixing in such cases as they can jump through bigger regions of the state space in one step by design. However, it does not appear that there

is a more in depth theoretical exploration of this point in the literature and so far we have mainly empirical evidence to support such claims.

Consider the following generative HDP mixture model with exponential family likelihood and conjugate prior over the component parameters:

$$\begin{aligned} G_0 \sim & \quad \mathrm{DP}(M_0, H) \\ G_j \sim & \quad \mathrm{DP}(N_0, G_0) \\ \theta_k \sim & \quad H \\ x_{j,i} \sim & F\left(\theta_{z_{j,c}^{global}} \left| z_{j,i}^{local} = c\right.\right) \end{aligned} \tag{4.9}$$

where H and F are again conjugate pairs of exponential family distributions. The probability of a point given the component parameter with which it is associated can be computed in exactly the same way as in 2.11 Chapter 3; similarly, the update for the shared component parameters θ_k given all points assigned to it would be unchanged.

Alternatively, the component parameters θ can also be integrated out and $p\left(x_{j,i} \left| \theta_{z_{j,c}^{global}}, z_{j,i}^{local} = c\right.\right)$ would be replaced with $p\left(x_{j,i} \left| \tau_{z_{j,c}^{global}}^{-ji}, \eta_{z_{j,c}^{global}}^{-ji}, z_{j,i}^{local} = c\right.\right)$ as in 3.14 in Chapter 3.

Algorithm 4.1 CRF-based Gibbs sampler for HDP

Input: $x_{j,1}, \ldots, x_{j,N_j}$ for all $j \in \{1, \ldots, J\}$: D-dimensional data; N_0: prior local count; M_0: prior global count;(τ_0, η_0): prior component hyperparameters
Output: Posterior of indicators: z^{local} and z^{global}; Posterior of component parameters:$(\theta_1, \ldots, \theta_K)$
Initialize $z_{j,i}^{local} = 1$ and $z_{j,1}^{global} = 1$ for all $j \in \{1, \ldots, J\}$ and $i \in \{1, \ldots, C_j\}$
$E_{\text{new}} = \infty$
repeat
  For $j \in \{1, \ldots J\}$;
    For $i \in \{1, \ldots, N_j\}$;
      For $c \in \{1, \ldots C_j\}$
        $d_{j,i,c} = \begin{cases} N_{j,c}^{-ji} p\left(x_{j,i} \left| \theta_k \right.\right) & \text{for existing local } c \\ \frac{N_0}{\sum_{k=1}^{K} M_k + M_0} \sum_{k=1}^{K} M_k p\left(x_{j,i} \left| \theta_k \right.\right) + M_0 p\left(x_{j,i} \left| \tau_0, \eta_0 \right.\right) & \text{for new } c \text{ drawn from the global DP} \end{cases}$
        $z_{j,i}^{local} \sim \text{Categorical}\left(\frac{d_{j,i,1}}{\sum_c d_{j,i,c}}, \ldots, \frac{d_{j,i,C_{j+1}}}{\sum_c d_{j,i,c}}\right)$
        If $z_{j,i}^{local} = C_j + 1$
          Sample values for $z_{j,c}^{global}$(using the same step as in the box below)
          $C_j \to C_j + 1$
  For $j \in \{1, \ldots J\}$;
    For $c \in \{1, \ldots, C_j + 1\}$;

        For $k \in \{1, \ldots K + 1\}$
          $q_{j,c,k} = \begin{cases} M_{\cdot k}^{-jc} \prod_{i:z_{ji}=c} p\left(x_{ji} \left| \theta_k \right.\right) & \text{for existing component } k \\ M_0 \prod_{i:z_{ji}=c} p\left(x_{ji} \left| \tau_0, \eta_0 \right.\right) & \text{for a new component } K+1 \end{cases}$
        $z_{jc}^{global} \sim \text{Categorical}\left(\frac{q_{j,c,1}}{\sum_k q_{j,c,k}}, \ldots, \frac{q_{j,c,K+1}}{\sum_k q_{j,c,k}}\right)$
        If $z_{jc}^{global} = K + 1$
          $\theta_{K+1} \sim H\left(\tau_0, \eta_0\right)$
          $K = K + 1$

  For $k \in \{1, \ldots, K\}$
    $\theta_k \sim H\left(\tau_k, \eta_k\right)$
until $(E_{\text{old}} - E_{\text{new}}) \to$ random

In order to update the local indicators z^{local}, for each dataset j we sweep through the observations in

that data set and update:

$$p\left(z_{j,i}^{local}=c\left|z_{-j,i}^{local},z_{j,c}^{global}\right.\right)\propto\begin{cases}N_{j,c}^{-ji}p\left(x_{j,i}\left|\theta_{z_{j,c}^{global}}\right.\right) & \text{for previous occupied } c\\ \frac{N_0}{\sum_{k=1}^{K}M_k+M_0}\left(\sum_{k=1}^{K}M_k p\left(x_{j,i}\left|\theta_k\right.\right)+M_0 p\left(x_{j,i}\left|\theta_0\right.\right)\right) & \text{for new } c\end{cases} \tag{4.10}$$

for all $i=1,\dots N_j$ with $N_j$denoting number of points in set j. If choose a new local c, we also have to sample its global indicator $z_{j,c}^{global}$. The global indicators z^{global} are updated for each j and c:

$$p\left(z_{j,c}^{global}=k\left|z_{-j,c}^{global},z_{j}^{local}=c\right.\right)\propto\begin{cases}M_k^{-jc}\prod_{i:z_{j,i}^{local}=c}p\left(x_{j,i}\left|\theta_k\right.\right) & \text{for existing } k\\ M_0\prod_{i:z_{j,i}^{local}=c}p\left(x_{j,i}\left|\theta_0\right.\right) & \text{for new } k\end{cases} \tag{4.11}$$

where note that to update indicator z_{jc}^{global} we compute the joint likelihood of all points associated to c (seated on table c) from dataset j. This is because a change in the global indicator $z_{j,c}^{global}$ changes the component allocation (this is essentially the top level allocation) of all points from c. The superscript $-jc$ in M_k^{-jc} denotes that we are excluding table c from set j when counting M_k. We update the component parameters $\theta_1,\dots,\theta_K$ in the same way as in the CRP-based Gibbs sampler where to find the points associated with component k we take all i and j s.t. $\left\{z_{j,i}^{local}=c\wedge z_{j,c}^{global}=k\right\}$. This representation involves $N_j\times J$ local indicators and $\sum_{j=1}^{J}C_j$ global ones. We outline the CRF-based Gibbs sampler in Algorithm 4.1.

Sampling with direct assignment

Integrating out the base measure G_0 in the CRF-based samplers introduces dependencies among the group-specific local indicators. This coupling complicates the inference, increases the required memory and can often slow down mixing. To mitigate those risks Teh *et al.* (2006) proposed two alternative Gibbs sampling procedures relying on alternative representations of the HDP: a *posterior sampling scheme with an augmented* representation and a *direct assignment* representation. The two are nearly equivalent and they use the stick-breaking construction of G_0 to avoid integrating over it. In addition, the direct assignment representation simplifies the bookkeeping by replacing the pair of local and global indicators with a single set of better-formulated component indicators. In this thesis we detail only the direct assignment representation, for the closely related augmented representation we refer readers to Teh *et al.* (2006).

Instead of inferring z^{local} and z^{global}, we can introduce a set of indicator variables $z_{j,i}$ which specify the component assignment of a point i from set j (for all $i\in\{1,\dots,N_j\}$ and $j\in\{1,\dots,J\}$). Grouping together the terms associated with each component k, we can compute the new assignment probabilities as:

$$p\left(z_{j,i}=k\left|z^{-ji},\beta\right.\right)\propto\begin{cases}\left(N_{j,k}^{-ji}+N_0\beta_k\right)p\left(x_{j,i}\left|\theta_k\right.\right) & \text{for existing } k\\ N_0\beta_{K+1}p\left(x_{j,i}\left|\theta_0\right.\right) & \text{for new } k=K+1\end{cases} \tag{4.12}$$

where now the counts $N_{j,k}$ denote number of points assigned to component k, $N_{j,k}=\sum_{i:z_{j,i}=k}1$; β are the mixing parameters defining the base measure G_0. From (4.2) and the definition of DP (see Equation (2.37)) we can write the distribution of the mixing parameters β as:

$$(\beta_1,\dots,\beta_K,\beta_{K+1})\sim\text{Dirichlet}\left(M_1,\dots,M_K,M_0\right) \tag{4.13}$$

The counts M_k were computed above using $M_k=\sum_{c:z_{j,c}^{global}=k}1$ as they reflect the number of different c across all datasets assigned to each of the shared components. In the new notation, explicit sampling of the

indicators z^{global} is avoided and instead we derive the relationship between the counts M and the assignments z. We cannot express M in a deterministic way through z, but we can use the component indicators to define a distribution over M. That is, in the direct assignment representation $z_{j,i}$ and M are sampled instead in place of $z_{j,i}^{local}$ and $z_{j,c}^{global}$. Recall that in the CRF construction, z^{local} and z^{global} were updated using (4.7) and (4.8), hence the counts M would increase every time an observation is assigned to an unused local cluster in (4.7). This is because M counts the assignments of local clusters to global ones and those are changed only when the conditional from (4.8) needs to be invoked.

Now let us express the distribution of the component indicators z in terms of z^{local} and z^{global}:

$$p\left(z_{j,i}=k\left|z_{-j,i}\right.\right)=\sum_{c=1}^{C_j+1} p\left(z_{j,i}^{local}=c\left|z_{-j,i}^{local}\right.\right) p\left(z_{j,c}^{global}=k\left|z_{-j,c}^{global}\right.\right) \tag{4.14}$$

by marginalizing the intermediate grouping of points in tables c. Using (4.14) we can also compute separately the probability of a point being assigned to an existing k through an existing intermediary c and the probability assignment to k was sampled using a new c:

$$p\left(z_{j,i}=k\left|z_{-j,i}\right.\right) \propto \begin{cases} N_{j,k} & \text{for used local cluster} \\ N_0\beta_k & \text{for unused local cluster} \end{cases} \tag{4.15}$$

Note that we have omitted any likelihood terms as at this stage we are interested only in the behaviour of the newly defined CRP.

The counts M_k can be computed using (4.15) by sequentially recording the number of times a new c would be needed (specific values of c, z^{local} and z^{global} are not actually needed). The way to do this by sampling from the Polya urn scheme defined in (4.15) $N_{j,k}$ times: we start with a single point and use (4.15) to sample its allocation; we update (4.15) accordingly to include the effect of that point; we add another point and sample its allocation according to the updated (4.15); each time allocation through an unused c is chosen (which always occurs with probability $N_0\beta_k$) we add a count to $N_{j,k}$ as it combines the reinforcement effect of all c's already sampled. We repeat this until $N_{j,k}$ points have been added and we count how many times allocation to an unused c was chosen. This count can be denoted with $M_{j,k}$ because in fact using this definition of $M_{j,k}$ implies $M_k=\sum_{j=1}^{J} M_{j,k}$. This is a Polya urn scheme[1] for sampling the counts $M_{j,k}$ which otherwise have a complex posterior distribution. (Van Gael, 2012)[2] discusses in more details the equivalents of the stationary distributions of samplers that treat z and M as random instead of z^{local} and z^{global}.

Instead of using a Polya urn scheme to compute $M_{j,k}$, we can also equivalently sample it directly from its posterior:

$$p\left(M_{j,k}=m\left|z,M_{-j,k},\beta\right.\right)=\frac{\Gamma\left(N_0\beta_k\right)}{\Gamma\left(N_0\beta_k+N_{j,k}\right)} s\left(N_{j,k},m\right)\left(N_0\beta_k\right)^m \tag{4.16}$$

which Antoniak (1974) has derived; $s\left(\cdot,\cdot\right)$ denotes unsigned *Stirling numbers* of first kind and $M_{j,k}$ is bounded between 1 and $N_{j,k}$. However, the direct approach of sampling the counts using (4.16) can often be challenging because in practice it is hard to evaluate Stirling numbers for even moderately large $N_{j,k}$ and m. This is why use the Polya-urn scheme for updating $M_{j,k}$ in the later experiments.

The Gibbs sampler for the HDP mixture model which uses the direct assignment construction of the

[1] (Blackwell, 1947) has shown that Polya urn schemes can be used to generate samples of random variables for which the posterior is either not available in direct form (for example this is the case for infinite mixture distributions), or it is not efficient to compute.

[2] (Van Gael, 2012) gives credit for the Polya urn scheme for sampling the counts M_{jk} to Emily Fox.

HDP iterates between: updates for z using (4.12); updates for the counts M; updates for the mixing weights $\beta_1, \ldots, \beta_{K+1}$ using (4.13) and updates of the component parameters $\theta_1, \ldots, \theta_K$ if they have not been integrated out from the corresponding exponential family posterior. We describe this Gibbs sampler in Algorithm 4.2.

Algorithm 4.2 Direct assignment Gibbs sampler for the HDP

Input: $x_{j,1}, \ldots, x_{j,N_j}$: for all $j \in \{1, \ldots, J\}$ D-dimensional data; N_0: prior local count; M_0: prior global count;(τ_0, η_0): prior component hyperparameters
Output: Posterior of indicators: z ; Posterior of component parameters:$(\theta_1, \ldots, \theta_K)$
Initialize $z_{j,i} = 1$ for all $j \in \{1, \ldots, J\}$ and $i \in \{1, \ldots, C_j\}$
$E_{\text{new}} = \infty$
repeat
  For $j \in \{1, \ldots J\}$;
    For $i \in \{1, \ldots, N_j\}$;

$$d_{j,i,c} = \begin{cases} \left(N_{j,c}^{-ji} + N_0 \beta_k\right) p\left(x_{j,i} \,|\theta_k\right) & \text{for existing } k \\ N_0 \beta_{K+1} p\left(x_{j,i} \,|\tau_0, \eta_0\right) & \text{for new } k = K+1 \end{cases}$$

$$z_{ji} \sim \text{Categorical}\left(\frac{d_{j,i,1}}{\sum_k d_{j,i,k}}, \ldots, \frac{d_{j,i,K+1}}{\sum_c d_{j,i,k}}\right)$$

      If $z_{ji} = K+1$
        $\theta_{K+1} \sim H\left(\tau_0, \eta_0\right)$
        $\nu \sim \text{Beta}\left(1, M_0\right)$
        $\beta_{K+2} = \beta_{K+1}\left(1 - \nu\right)$
        $\beta_{K+1} = \beta_{K+1}\nu$
        $K \to K+1$
  For $j \in \{1, \ldots J\}$;
    For $k \in \{1, \ldots K+1\}$
      Sample $M_{j,k}$ using either (4.16) or successively (4.15) as described in the text
  For $k \in \{1, \ldots, K\}$
    Compute $M_k = \sum_{j=1}^{J} M_{j,k}$
    $\theta_k \sim H\left(\tau_k, \eta_k\right)$
  $\beta_1, \ldots, \beta_K, \beta_{K+1} \sim \text{Dirichlet}\left(M_1, \ldots, M_K, M_0\right)$
until $(E_{\text{old}} - E_{\text{new}}) \to$ random

4.4 Deterministic inference for HDPs

To date, most deterministic methods that have been developed for inference in HDP are different variants of VB inference methods. Liang *et al.* (2007) first suggested VB inference for a non-collapsed representation of the HDP and suggested point estimate approximations for the mixing parameters β. Later Teh *et al.* (2007) introduced a VB method derived for the collapsed HDP relying on the CRF construction. However, this method is based on a sophisticated family of marginal likelihood bounds and so it leads to challenging optimization and sensitivity to initialization. Wang *et al.* (2011) introduced an online VB inference method also based on the CRF construction which is able to efficiently process significantly larger datasets. This is done by optimizing the variational bound only using random subsets of the data, one at a time. This scheme replaces the usual coordinate ascent (Wright, 2015) from the traditional VB with the *natural gradient* (Amari, 1998) leading to efficient parallelization. Bryant & Sudderth (2012) has presented a more robust online VB method that relies on the direct assignment representation of the HDP and makes use of split and merge moves to avoid poor local minima.

Wang *et al.* (2011) suggested a somewhat different VB approach that builds upon the more typical VB method (Blei & Jordan, 2006) by adding stochastic Gibbs split and merge moves to avoid the implicit trun-

cation of the variational distrbution of non-collapsed Bayesian nonparametrics models. While theoretically convenient, the Gibbs steps can lead to significantly slower mixing. In contrast to streaming (online) VB methods, Hughes *et al.* (2015a) recently suggested a VB method that optimizes a mean field objective function with coordinate descent algorithm in a typical fashion. However, this is done in atypical order where only some of the variational parameters are updated at a time and data is processed one batch at a time.

This method introduces a novel surrogate likelihood bound that allows the definition of full variational posteriors over each of the random variables in direct assignment HDP mixtures. Repeated sweeps through the full data are required, but the whole dataset does not have to be stored in memory as Hughes *et al.* (2015a) suggests keeping efficient *memoized statistics* that summarize previously processed batches. This method further benefits from efficient split and merge moves that allow the iterations to escape poor local solutions; scalability and efficient memory representation strongly depend on the number of batches in which the full dataset has been split.

In this section we discuss in detail some simpler alternatives for scaling up inference in the HDP mixture model. We first review Jiang *et al.* (2013) which extends the SVA formalism to the more complex case of hierarchical clustering. A simple scalable algorithm is derived from the CRF construction of the HDP by scaling the component variances. In contrast to Jiang *et al.* (2013) we introduce an iterative MAP approach for inference in HDP mixtures which overcomes many of the issues implicit to SVA methods for hierarchical models; in Section 4.5 we empirically evaluate both methods.

4.4.1 SVA inference for HDP mixtures

As with SVA inference for the DPMM, Jiang *et al.* (2013) have extended the SVA approach for inference in HDP mixtures and we review its derivation. We will start from the CRF construction of the HDP and assume an exponential family likelihood $\tilde{F}\left(\tilde{\theta}\right)$ (see (4.9)) which is parametrized by a scaled natural parameter $\tilde{\theta} = \xi\theta$ and log-partition function $\tilde{\psi}\left(\tilde{\theta}\right) = \xi\psi\left(\tilde{\theta}/\xi\right)$ for some $\xi > 0$. The prior hyperparameters τ and η are also scaled giving $\tilde{\tau} = \frac{\tau}{\xi}$ and $\tilde{\eta} = \frac{\eta}{\xi}$. Finally the concentration parameters N_0 and M_0 are replaced with some threshold parameters λ_1 and λ_2 where we assume the relationship $N_0 = \left(g_{\tilde{\phi}}\left(\frac{\boldsymbol{\tau}}{\xi}, \frac{\eta}{\xi}\right)\left(\frac{2\pi}{\xi+\eta}\right)^{D/2}\xi^D\right)^{-1}\exp\left(-\xi\lambda_1\right)$ and $M_0 = \left(g_{\tilde{\phi}}\left(\frac{\tau}{\xi}, \frac{\eta}{\xi}\right)\left(\frac{2\pi}{\xi+\eta}\right)^{D/2}\xi^D\right)^{-1}\exp\left(-\xi\lambda_2\right)$. Then in taking the limit $\xi \to \infty$ of the scaling parameter we can derive a hierarchical clustering procedure that approximately minimizes the following objective function:

$$\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i:z_{j,i}=k} D_\phi\left(x_{j,i}, \mu_k\right) + \lambda_1\sum_{j=1}^{J} C_j + \lambda_2 K \tag{4.17}$$

with respect to the corresponding indicator variables and component means. In each dataset j points are clustered around exactly C_j of the overall K centroids; those C_j clusters are local clusters to dataset j. λ_1 acts as a local threshold parameter controlling the number C_j available in each dataset and λ_2 acts as a global threshold parameter controlling the number of centroids K that are instantiated (smaller λ_1 and λ_2 leads to larger C_j and K).

Point updates

For an observation $x_{j,i}$ the resulting SVA algorithm computes K distances $D_\phi\left(x_{j,i}, \mu_1\right), \ldots, D_\phi\left(x_{j,i}, \mu_K\right)$ to each centroid. For each μ_k which is not yet used in dataset j, we add a penalty λ_1 to that distance. We compare all of the distances and assign $x_{j,i}$ to the closest centroid μ_k unless the distance to it is larger than

the penalty $\lambda_1 + \lambda_2$. If even the smallest of the k distances is larger than $\lambda_1 + \lambda_2$, a new centroid centered at $x_{j,i}$ is created; observation $x_{j,i}$ gets assigned to the new centroid ($z_{j,i} = K + 1$); the centroid μ_{K+1} is added to the list of available ones for dataset j, and C_j and K are increased by 1 accordingly. This step is repeated for all points i in all datasets.

Local updates

Next, for each dataset j we sweep through local clusters c ($c = 1, \ldots, C_j$) and update their association with centroids $\mu_1, \ldots, \mu_K$. Essentially the association of local clusters to centroids determines which of the C_j centroids represented in set j. For each local cluster, we compute the sum of distances between its data point and centroid μ_k. This is repeated for all $k \in \{1, \ldots, K\}$ giving us K sums to evaluate. We also evaluate a $K + 1$ term which is λ_2 added to the sum of all distances of points in local cluster c from their own mean. To summarize, the $K + 1$ terms we compare are:

$$\left(\sum_{i:z_{j,i}=1} D_\phi \left(x_{j,i}, \mu_1 \right), \ldots, \sum_{i:z_{j,i}=K} D_\phi \left(x_{j,i}, \mu_K \right), \lambda_2 + \sum_{x \in S_{jc}} D_\phi \left(x_{j,i}, \bar{x}_{j,c} \right) \right) \tag{4.18}$$

with $\bar{x}_{j,k}$ denoting the mean of points associated to c from set j (points in the same local cluster) and S_{jc} denoting the set of observations in local cluster c from set j. We choose the smallest of those terms: if it is one of the first K terms we assign all points from S_{jc} to the corresponding centroid; if it is the last term we create a new centroid $\mu_{K+1} = \bar{x}_{j,c}$.

Global updates

The centroids μ_k are updated by computing the average of all points across all sets which are assigned to cluster k.

This SVA-HDP algorithm for clustering multiple batches of data has all the drawbacks that the SVA algorithm from Section (3.7.2) has. By replacing the prior counts N_0 and M_0 with the thresholds λ_1 and λ_2 we lose interpretability of the model hyperparameters, we lose control over some of the hyperparameters and we also assume arbitrary connections in the underlying probabilistic model. When we scale τ and η with an equal factor ξ and further take the limit $\xi \to \infty$, we effectively shrink all of the component covariaces to 0 and assume points are spread equally among each of the clusters across all dimensions. In addition, the reinforcement terms coming from the counts N (lower level DPs) and M (top level DP) have no effect, therefore data is grouped based on the geometry of the data space alone. This implies the assumption that clusters are not only inherently spherical but also contain an equal number of observations. The asymptotic assumptions also make the model likelihood degenerate which prohibit us from using standard model selection and prediction techniques.

The SVA algorithm for the special case of the Gaussian HDP mixture model was first introduced in Kulis & Jordan (2011). Kulis & Jordan (2011) called the method the *HDP-means* algorithm and it can be seen as a special case of SVA-HDP where the Bregman divergence $D_\phi(\cdot)$ is replaced with squared Euclidean distance $\|\cdot\|_2^2$ as a measure of closeness of points to their centroids.

In the next section we propose an iterative MAP inference algorithm that mitigates all of those drawbacks while being as simple and as fast.

4.4.2 Iterative maximum a-posteriori inference

Based on the different constructions of the HDP mixture model, we can define several different iterative MAP algorithms for fast inference in the model. Despite all being derived from the same probabilistic graphical model, those algorithms are likely to tolerate different local optima, to converge at different rates and to have different memory requirements. In this section we will introduce a MAP-based algorithm derived using the CRF construction of the underlying HDP which we call *MAP-HDP*.

This is the same construction used for the SVA approach in Section 4.4.1 therefore we can make direct comparisons. In Chapter 5 we also introduce a MAP-based method which uses the direct assignment construction of the HDP, however the method in Chapter 5 is used for inference in HDPs for sequential modeling: the HDP-HMM. We will not discuss in detail the different algorithms which may or may not integrate over the component parameters θ because a lot of the issues are almost identical to those raised in Chapter 3.

As with the related CRF-based Gibbs sampler from Section 4.3.3, MAP-HDP iterates between updates for the local indicators z^{local}, the global indicators z^{global} and the component parameters θ (if they have not been integrated out). The indicators are updated one at a time with the value maximizing their corresponding posterior. Those updates can be easily derived using the expressions in (4.10) and (4.11). We summarize the MAP-HDP in Algorithm 4.3. The proposed method converges to a fixed point local maxima of the complete data likelihood:

$$p\left(x, z^{local}, z^{global} \left| \theta \right.\right) = \prod_{j=1}^{J} \prod_{i=1}^{N_{j\cdot}} \prod_{c=1}^{C_j} \prod_{k=1}^{K} p\left(x_{j,i} \left| \theta_k \right.\right)^{\delta_{z,k}} p\left(z^{local}, z^{global}\right) \tag{4.19}$$

where for each j, $p\left(x_{j,i} \left| \theta_k \right.\right)$ is computed in the same way as $p\left(x_i \left| \theta_k \right.\right)$ in earlier chapters and we have used the shorter notation $\delta_{z,k} = \delta_{z_{ji}^{local},c}\delta_{z_{jc}^{global},k}$. The joint distribution over the indicator variables $p\left(z^{local}, z^{global}\right)$ is the probability of partitions induced by the CRF:

$$p\left(z^{local}, z^{global}\right) = \frac{M_0^{K+1}\Gamma\left(M_0+1\right)}{\Gamma\left(M_0+M_{j,k}\right)} \prod_{k=1}^{K} M_k \prod_{j=1}^{J} \left(\frac{N_0^{M_{j,\cdot}}\Gamma\left(N_0+1\right)}{\Gamma\left(N_0+N_{j,\cdot}\right)} \prod_{k=1}^{K} \prod_{c:z_{jc}=k} N_{j,c}! \right) \tag{4.20}$$

where $M_{j,\cdot} = \sum_{k=1}^{K} M_{j,k}$. Typically, it will be easier to compute the log-likelihood of the expression in (4.19). An efficient trick to assess convergence in MAP-HDP can also be to compute $\sum_{j=1}^{J}\sum_{i=1}^{N_{j,\cdot}}\sum_{c=1}^{C_j} d_{j,i,c} + \sum_{j=1}^{J}\sum_{c=1}^{C_j}\sum_{k=1}^{K} q_{j,c,k}$ from Algorithm 4.3 and stop the algorithm when this sum stops changing.

Out-of-sample prediction

As with Section 3.8.3 for the DPMM we consider two ways of computing out-of-sample likelihoods commonly used for prediction or model selection:

- *Mixture predictive density.* If we integrate over the local and global indicators, we obtain the following predictive mixture density:

$$p\left(x_{j,N_j+1} \left| z^{local}, z^{global}, \theta \right.\right) = \sum_{c=1}^{C_j+1} p\left(z_{j,i}^{local} = c \left| z_{-j,i}^{local} \right.\right) \sum_{k=1}^{K+1} p\left(z_{j,c}^{global} = k \left| z_{j,i}^{local} = c, z_{-j,c}^{global} \right.\right) p\left(x_{N+1} \left| \theta_k \right.\right) p\left(\theta_0\right) \tag{4.21}$$

  where we have omitted the conditioning on the training data X; the local assignment probability $p\left(z_{j,i}^{local} = c \left| z_{-j,i}^{local} \right.\right)$ is computed from the local CRP defined in (4.7) and $p\left(z_{j,c}^{global} = k \left| z_{j,i}^{local} = c, z_{-j,c}^{global} \right.\right)$ is computed from the global CRP defined in (4.8).

Algorithm 4.3 MAP-HDP algorithm

Input: $x_{j,1},\ldots,x_{j,N_j}$: for all $j \in \{1,\ldots,J\}$ D-dimensional data; N_0: prior local count; M_0: prior global count;(τ_0,η_0): prior component hyperparameters; ϵ: convergence threshold;
Output: Local level indicators: z^{local}; Top level indicators z^{global}; Point estimates for the parameters:$(\theta_1,\ldots,\theta_K)$
Initialize $z_{j,i}^{local}=1$ and $z_{j,1}^{global}=1$ for all $j \in \{1,\ldots,J\}$ and $i \in \{1,\ldots,N_j\}$
$E_{\text{new}}=\infty$
repeat
  For $j \in \{1,\ldots J\}$
    For $i \in \{1,\ldots,N_j\}$
      For $c \in \{1,\ldots C_j\}$

$$d_{j,i,c}=\begin{cases}-\ln N_{j,c}^{-ji}-\ln p\left(x_{j,i}\left|\theta_{z_{j,c}^{global}}\right.\right) & \text{for existing } c \text{ in set } j\\ -\ln\left(\frac{N_0}{\sum_{k=1}^{K}M_k+M_0-K}\right)-\ln\left(\sum_{k=1}^{K}M_k p\left(x_{j,i}\left|\theta_k\right.\right)+M_0 p\left(x_{j,i}\left|\tau_0,\eta_0\right.\right)\right) & \text{for a new } c\end{cases}$$

      $z_{ji}^{local}=\arg\min_{c\in\{1,\ldots,C_j+1\}} d_{j,i,c}$
      If $z_{ji}^{local}=C_j+1$
        Update values of $z_{j,c}^{global}$ (using the same step as in the box below)
        $C_j \to C_j+1$
        If $z_{j,c}^{global}=K+1$
          $\theta_{K+1}=\arg\max_\theta p\left(\theta\left|\tau_0+x_{j,i},\eta_0+1\right.\right)$
          $K=K+1$
  For $j \in \{1,\ldots J\}$;
    For $c \in \{1,\ldots,C_j+1\}$;

      For $k \in \{1,\ldots K+1\}$

$$q_{j,c,k}=\begin{cases}-\ln M_{\cdot k}^{-jc}-\sum_{i:z_{ji}^{local}=c}\ln p\left(x_{ji}\left|\theta_k\right.\right) & \text{for existing component } k\\ -\ln M_0-\sum_{i:z_{ji}=c}\ln p\left(x_{ji}\left|\tau_0,\eta_0\right.\right) & \text{for new } k=K+1\end{cases}$$

      $z_{jc}^{global}=\arg\min_{k\in\{1,\ldots,K+1\}} q_{j,c,k}$
      If $z_{jc}^{global}=K+1$
        $\theta_{K+1}=\arg\max_\theta p\left(\theta\left|\tau_0+x_i,\eta_0+1\right.\right)$
        $K=K+1$
  For $k \in \{1,\ldots,K\}$

$$\theta_k=\arg\max_\theta p\left(\theta\left|\tau_0+\sum_{i,j:z_{j,i}^{local}=c,z_{j,c}^{global}=k} g\left(x_{j,i}\right),\eta_0+\sum_{i,j:z_{j,i}^{local}=c,z_{j,c}^{global}=k}1\right.\right)$$

until $(E_{\text{old}}-E_{\text{new}})<\epsilon$

- *MAP cluster assignments.* Alternatively, we can use point estimates for the indicator variables z_{j,N_j+1}^{local} and $z_{j,z_{j,N_j+1}^{local}}^{global}$. We first estimate z_{j,N_j+1}^{local} by choosing the value that minimizes its negative log posterior:

$$\left(z_{j,N_j+1}^{local}\right)^{MAP}=\underset{c\in\{1,\ldots,C_j+1\}}{\arg\min}\left[-\ln p\left(x_{j,N_j+1}\left|\theta_{z_{j,c}^{global}}\right.\right)-\ln p\left(z_{j,N_j+1}^{local}=c\left|z_{-j,i}^{local}\right.\right)\right]. \tag{4.22}$$

  Then, we use this value to estimate $z_{j,z_{j,N_j+1}^{local}}^{global}$ by minimizing its negative log posterior conditioned on the found $\left(z_{j,N_j+1}^{local}\right)^{MAP}$:

$$\left(z_{jz_{j,N_j+1}}^{global}\right)^{MAP}=\underset{k\in\{1,\ldots,K+1\}}{\arg\min}\left[-\ln p\left(x_{j,N_j+1}\left|\theta_k\right.\right)-\ln p\left(z_{j,z_{j,N_j+1}}=k\left|\left(z_{j,N_j+1}^{local}\right)^{MAP},z_{-j,c}^{global}\right.\right)\right] \tag{4.23}$$

As with the DPMM case in Section 3.8.3, the first approach will be more robust because it incorporates the effect of all mixture components. The fact that the predictive mixture density is computed only using point estimates of the indicator variables of the training data leads to an approximation error which does

Table 4.1: NMI score for MAP and SVA methods for HDP mixtures applied to Gaussian data.

Data set	HDP-means	MAP-HDP
Spherical	0.81	0.84
Elliptical	0.44	0.77

not arise for out-of-sample predictions in MCMC. This error often leads to underestimated variance in the predictions. At the same time, the SVA method from Section 4.4.1 does not provide even an approximate way to compute prediction probabilities of new points. Therefore, despite being extremely scalable and deterministic, MAP-HDP is model-based and allows for rigorous model selection and prediction.

4.5 Synthetic study

In this section we compare the MAP-HDP and HDP-means algorithm for the case of spherical and elliptical synthetic Gaussian data. To generate the spherical synthetic data, we start by sampling the component parameters for 15 Gaussian distributions: the component means are sampled uniformly in $[0, 1]^2$; the component covariances are assumed to be fixed at $(.01\mathbf{I})$. We generate 50 datasets by choosing at random 5 of the 15 Gaussian components and sampling 5 points from each of the chosen components (in Figure 4.3 we show four of those sets)[3]. To generate the elliptical data we simply fix the component covariances to be $\left[10^{-6}, 10^{-2}\right]$ instead of $(.01\mathbf{I})$.

The parameters for the HDP-means algorithm were set such that the resulting clustering has the correct number of global and local clusters (as pointed by Kulis & Jordan (2011)). The prior hyperparameters for the MAP-HDP are set in the same way as we did for the MAP-DP algorithm (see Appendix E). In the case of the spherical data we derive MAP-HDP from an HDP mixture model with spherical Gaussian components. For the case of elliptical data, we start from an HDP mixture with elliptical Gaussian components (with a univariate Normal-Gamma prior placed over each dimension of the component means and variances). If we wish to maximize MAP-HDP performance typically we should assume a model with full covariances.

In Table 4.1 we display the clustering accuracy of the methods in terms of NMI score between the estimated and the ground truth component assignments[4]. On the spherical data the performance of MAP-HDP and HDP-means is nearly identical. This can be explained by the fact that the data is not just spherical, but also equally distributed among the different components. Therefore, most of the restrictive assumptions of SVA algorithms are met in the experimental setup.

In the elliptical case we observe a marked deterioration in the clustering accuracy of HDP-means which is not the case for MAP-HDP. As for all SVA methods, departure from sphericity cannot be modeled adequately whereas iterative MAP just needs to be adapted to a more flexible probabilistic graphical model.

It is interesting to see if both HDP-means and MAP-HDP would actually outperform a non-hierarchical treatment of the same problem. For a fair comparison, consider only the spherical synthetic data. If we use K-means (with the true K) to cluster a single batch combining all 50 datasets, K-means scores NMI of 0.61; if we use K-means to cluster each of the datasets separately and take the average NMI, K-means scores an average NMI of 0.78.

[3]This setup is identical to the synthetic study in Kulis & Jordan (2011).

[4]The component assignments for each point are obtained by combining the local and global indicators. If $x_{j,i}$ is associated with $z_{j,i}^{local} = c$ and we have $z_{j,c}^{global} = k$, the component assignment of $x_{j,i}$ is k.

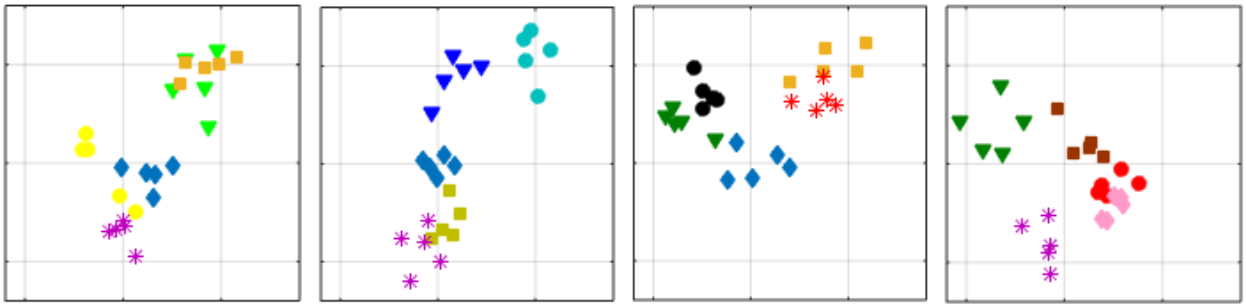


Figure 4.3: Four different synthetic data sets used for the HDP experiment explained in the text. Points denoted with the same shape and color across different plots are generated from the same underlying component.

4.6 Discussion

We have extended the iterative MAP approach to derive a simple deterministic inference framework for HDP-based models. We reviewed the different representations of the HDP and proposed an example MAP algorithm using the CRF contruction of the HDP. The proposed algorithm is contrasted to an SVA inference algorithm for the HDP and we observe that the many problems inherent to SVA methods also translate to more complex models such as the HDP. However, we demonstrate that SVA and MAP methods derived from the HDP benefit from the architecture of the underlying probabilistic model and outperform simpler methods such as K-means for the problem of modeling local and global structure in groups of related data.

Hierarchical Dirichlet processes play an important role as a building block for more elaborate probabilistic models. A particularly important example of that is the extension of HDP to sequential modeling which we discuss in Chapter 5. Efficient scalable methods such as MAP-HDP can be used for such extensions to the HDP to reduce the substantial computational requirements for inference in such deeply nested models.

Some of the practical models we plan to study in future include:

- The dynamic HDP in (Ren *et al.*, 2008) which extends the HDP to model time evolving data, where the set specific measures $G_1, \ldots, G_J$ are no longer assumed to be independent given the shared G_0. Each data set is assumed to be collected at a consecutive point in time and the measure for set j, G_j, is assumed to be strongly correlated to the previous and next measures G_{j-1} and $G_{j+1}$where this relationship is specified using the dynamic DP mixture in (Dunson, 2006).

- HDP with random effects described in (Kim & Smyth, 2006) which uses the HDP to construct a random effects model for multiple data sets.

- The nested HDP (Paisley *et al.*, 2012) originally proposed for modeling topics from multiple documents in a tree-like hierarchy.

Chapter 5

Model-based nonparametric analysis of sequential data

5.1 Introduction

Sequential data are at the core of many statistical modeling and machine learning problems. For example, text consists of sequences of words, financial data are often sequences of share prices, speech signals are sequences of acoustic pressure readings, proteins are sequences of amino acids, DNA are sequences of nucleotides and accelerometer data is a sequence of acceleration outputs measured at consecutive points in time. Although it is possible to directly model the relationships between subsequent elements of a time series, e.g. using auto-regressive or n-gram models, in some cases it is more useful to assume some underlying probabilistic structure and model it with a hierarchical model. For example, observed accelerometer outputs might correspond to human physical behaviours and latent variables can be used to allocate the behaviours to the raw data.

In the case of time-independent data, we already demonstrated how latent variable models such as mixture models can be designed and inferred. An adequate sequential model however, needs to incorporate some serial dependence between the latent variables. Consider a finite mixture model in which the indicator variables are assumed to be serially dependent, the resulting sequential model is known as a *hidden Markov model* (HMM)(Section 5.2). In mixture models each point (independent of time) is modeled with the same mixture distribution, but in HMMs the mixture distribution modeling the density of some point t depends on the indicator associated with time point $t-1$. In the parametric setting of the HMM, the indicator for point $t-1$ can take some fixed K values. Therefore, under the HMM any point from the data is modeled with 1 of K different mixture distributions. The coefficients for all those mixture distributions are written in a $K \times K$ table called a *transition matrix*. The component parameters for those mixture distributions are shared across time so there are only K of them. Such a model is known as a *stationary (homogeneous)* HMM.

Now consider we want to obtain a nonparametric version of the HMM. We cannot simply relax the assumption of fixed K and model the data with a collection of different DPMMs as the component parameters of different DPMMs are different with probability 1. Instead, Beal *et al.* (2002) proposed to model the collection of mixtures that form a HMM using a hierarchical Polya urn scheme later formalized by Teh *et al.* (2006) as the HDP. Similar to the way in which the DP was used as a nonparametric prior over the mixing measure to define infinite mixtures, the HDP can be used to describe a nonparametric prior of the HMM transition parameters in order to define an *infinite hidden Markov model* (iHMM)(Beal *et al.*, 2002; Teh *et al.*, 2006).

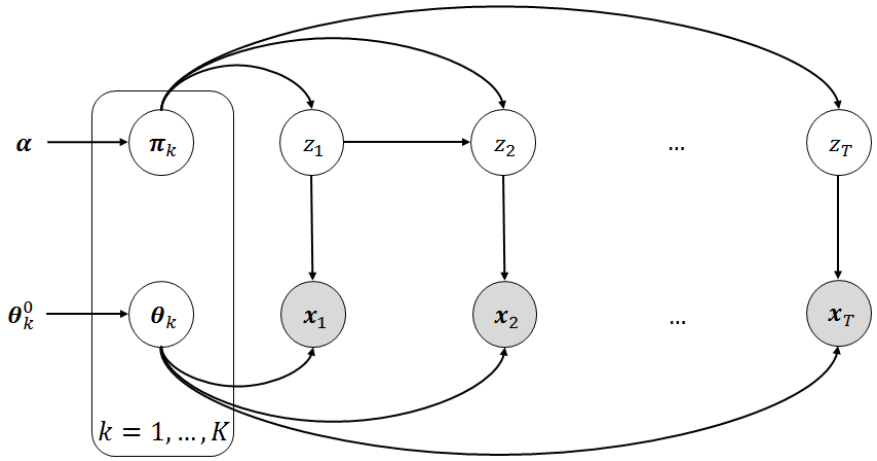


Figure 5.1: Graphical model for the Bayesian HMM.

In this chapter we formally define the HMM and focus in particular on its nonparametric extension and the issues specifically relevant to inference in this BNP discrete latent variable model for sequential data. In Section 5.4.2 we propose two novel iterative MAP methods for deterministic inference in the iHMM which we call *MAP-iHMM* and *dynamic MAP-iHMM* respectively. MAP-iHMM is derived using the direct assignment construction of the HDP underlying the transition parameters and it can be derived from the direct assignment Gibbs sampler for iHMM. Both MAP-iHMM and the direct assignment Gibbs sampler update the hidden states at each step conditioned on all other states in the Markov chain. This is a consequence of the coupling introduced when we integrate out the local DPs (see Section 4.3.3). However, typically the state indicator variables in HMMs are strongly correlated and it is a well known fact that the Gibbs sampler mixes slowly when there are strong correlations between variables. As a result, in practice methods which update the whole sequence of state indicators in a block are preferred for sequential models. For parametric HMMs widely used methods like this are *the forward-filtering backward-sampling* (FF-BS) and *Viterbi* algorithms (both defined in Section 5.2). In the case of BNP extensions of the HMMs these methods are not trivially extended: Van Gael *et al.* (2008) has proposed a variant of FF-BS for inference in the iHMM, and the dynamic MAP-iHMM we propose in Section 5.4.2 can be seen as a nonparametric extension of the Viterbi algorithm.

5.2 Hidden Markov models

The HMM (Rabiner & Juang, 1986) is a ubiquitous probabilistic model for analysis of sequential data. HMMs have been widely applied across many disciplines such as: speech recognition (Jelinek, 1997; Rabiner & Juang, 1993), natural language modeling (Manning & Schütze, 1999), online handwriting recognition (Nag *et al.*, 1986), for the analysis of biological sequences such as proteins and DNA (Krogh *et al.*, 1994; Durbin *et al.*, 1998; Baldi & Brunak, 2001) and also for modeling behavioural patterns of sequential data from a wide array of sources (Andrade *et al.*, 2006; Toreyin *et al.*, 2008; Chung & Liu, 2008; Gao *et al.*, 2006). Much like mixture models, the HMM describes the probability distribution over a sequence of observations $x_1, \ldots, x_T$ of some length T. Every point t is associated with a state indicator z_t, for $t = 1, \ldots, T$, and each state indicator points to one of K possible states (components). Given the indicator variables and the component parameters, the observations are independent and identically distributed, however the indicator variables are assumed to have the *Markov property*. In the case of *first order* chains, the distribution of the state indicator z_t at time t depends only on the state indicators immediately before it z_{t-1} and this dependence is characterized by a

$K \times K$ *stochastic* (*transition*) *matrix* π, where $\pi_{i,j} = p(z_t = j \,|z_{t-1} = i)$. In this work we focus on simple *stationary* Markov models where the transition matrix π is independent of time t. Given the state for time $t-1$, observation x_t is modeled with a K component mixture distribution with mixture weights defined by the row of the transition matrix pointed by z_{t-1}, that is $\pi_{z_{t-1},1}, \ldots, \pi_{z_{t-1},K}$; each of the K rows specify the weights for a different mixture model. Each mixture distribution shares the same component parameters (also called *emission* parameters) $\theta_1, \ldots, \theta_K$, which are independent of time (they are *homogeneous*). More formally we can write:

$$p(x_t \,|z_{t-1} = j, \theta) = \sum_{k=1}^{K} \pi_{j,k} p(x_t \,|\theta_k) \tag{5.1}$$

As with previous chapters in this thesis, we will focus on the cases where $p(x_t \,|\theta_k)$ is of the exponential family: for example in speech recognition we often use normally distributed components (Martin & Jurafsky, 2000) and in natural language processing we typically use multinomial components. For first order stationary HMMs we can write the complete data model likelihood for the HMM as:

$$p(x, z \,|\pi, \theta) = \prod_{t=1}^{T} p(z_t \,|z_{t-1}) \, p(x_t \,|z_t) = \prod_{t=1}^{T} \pi_{z_{t-1},z_t} p(x_t \,|\theta_{z_t}) \tag{5.2}$$

Three inference problems typically occur when modeling data with HMMs:

- Given π, θ and the data x, infer the distribution over the hidden variables $z_1, \ldots, z_T$. This computational problem is used in applications such as *filtering* and *smoothing*. Typically it is addressed using the *forward algorithm* or *forward-backward algorithm*.

- Given π, θ and the data x, infer the most likely sequence of states associated with the sequence of observations $x_1, \ldots, x_T$. This is a common problem in speech recognition, bioinformatics and many other applications which is usually solved using the *Viterbi algorithm* (Viterbi, 1967); this most likely sequence is also known as the *Viterbi path*.

- Inferring the model parameters π and θ and the distribution of the latent variables $z_1, \ldots, z_T$ using the sequence of observed data $x_1, \ldots, x_T$. This is the most general inference task related to HMMs and it is usually approached with approximate methods such as maximum likelihood estimation. An adaptation of the E-M algorithm for the HMM is called the *Baum-Welch algorithm* (Baum *et al.*, 1970), a type of *forward-backward* algorithm.

Bayesian analysis of the HMM involves placing priors over the unknown π and θ; $p(\pi)$ and $p(\theta)$. The graphical model of the Bayesian HMM is displayed in Figure 5.1. Typically we choose a conjugate Dirichlet prior over each row of the transition matrix: $\pi_{k,\cdot} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$ for $k = 1, \ldots, K$. As with the case of mixture models, a conjugate prior over the state parameters $\theta_1, \ldots, \theta_K$ is guaranteed to exist in closed form whenever the emission probabilities are from the exponential family. One simple way to do inference in the Bayesian HMM is by using collapsed Gibbs sampling schemes (Albert & Chib, 1993; Robert *et al.*, 1993), summarized in Algorithm **5.1**.

	Algorithm 5.1: Collapsed Gibbs for HMM (spherical Gaussian)		**Algorithm 5.2**: Forward-filtering, backward-sampling (spherical Gaussian)
Input	$x_1, \ldots, x_T$: D-dimensional data		$x_1, \ldots, x_T$: D-dimensional data
	K: number of clusters		K: number of clusters
	α: concentration parameter		α: concentration parameter
	σ: spherical cluster variance		σ: spherical cluster variance
	σ_0: prior centroid variance		σ_0: prior centroid variance
	μ_0: prior centroid variance		μ_0: prior centroid variance
Output	Posterior of indicators: $(z_1, \ldots, z_T)$		Posterior of indicators: $(z_1, \ldots, z_T)$
	Posterior of transition matrix: π		Posterior of transition matrix: π
	Posterior of :$(\mu_1, \ldots, \mu_K)$		Posterior of :$(\mu_1, \ldots, \mu_K)$
	Sample parameters from the prior		Sample parameters from the prior
1	$\mu_1, \ldots, \mu_K \overset{i.i.d.}{\sim} \mathcal{N}(\mu_0, \sigma_0)$	**1**	$\mu_1, \ldots, \mu_K \overset{i.i.d.}{\sim} \mathcal{N}(\mu_0, \sigma_0)$
2		**2**	$(\pi_{k,1}, \ldots, \pi_{k,K})_{k=1}^{K} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
3	$z_1, \ldots, z_T \sim$ DirMulti $\left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$	**3**	$z_1, \ldots, z_T \sim \text{DirMulti}\left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$
4	**repeat**	**4**	**repeat**
5		**5**	**for** $k \in 1, \ldots, K$
6		**6**	$d_{1,k} = \frac{1}{2\sigma}\|x_1 - \mu_k\|_2^2 + \frac{D}{2}\ln\sigma - \ln\sum_i \pi_{k,i}$
7	**for** $t \in 1, \ldots, T$	**7**	**for** $t \in 2, \ldots, T$
8	**for** $k \in 1, \ldots, K$	**8**	**for** $k \in 1, \ldots, K$
9	$\dot{\pi}_{z_{t-1},k} = N^{-t}_{z_{t-1},k} + \alpha/K;$ $\dot{\pi}_{k,z_{t+1}} = N^{-t}_{k,z_{t+1}} + \alpha/K;$	**9**	
10	$d_{t,k} = \frac{1}{2\sigma}\|x_t - \mu_k\|_2^2 + \frac{D}{2}\ln\sigma - \ln\dot{\pi}_{z_{t-1},k} - \ln\dot{\pi}_{k,z_{t+1}}$	**10**	$d_{t,k} = \frac{1}{2\sigma}\|x_t - \mu_k\|_2^2 + \frac{D}{2}\ln\sigma - \ln\sum_i \pi_{k,i} + d_{t-1,z_{t-1}}$
11	$d_{t,k} = \exp(-d_{t,k})$	**11**	$d_{t,k} = \exp(-d_{t,k})$
12	$z_t \sim \text{Categorical}\left(\frac{d_{t,1}}{\sum_k d_{t,k}}, \ldots, \frac{d_{t,K}}{\sum_k d_{t,k}}\right)$	**12**	$z_T \sim \text{Categorical}\left(\frac{d_{T,1}}{\sum_k d_{T,k}}, \ldots, \frac{d_{T,K}}{\sum_k d_{T,k}}\right)$
13		**13**	**for** $t \in T-1, \ldots, 1$
14		**14**	**for** $k \in 1, \ldots, K$
135		**15**	$q_k = \frac{d_{t,k}}{\sum_{i=1}^{K} d_{t,i}} \pi_{z_t, z_{t+1}}$
16		**16**	$z_t \sim \text{Categorical}(q_1, \ldots, q_K)$
17	**for** $k \in 1, \ldots, K$	**17**	**for** $k \in 1, \ldots, K$
18	$\dot{\sigma}_k = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma}N_k\right)^{-1}$	**18**	$\dot{\sigma}_k = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma}N_k\right)^{-1}$
19	$\dot{\mu}_k = \dot{\sigma}_k\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\bar{x}_k\right)$	**19**	$\dot{\mu}_k = \dot{\sigma}_k\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\bar{x}_k\right)$
20		**120**	$(\dot{\pi}_{k,1}, \ldots, \dot{\pi}_{k,K}) = \left(N_{k,1} + \frac{\alpha}{K}, \ldots, N_{k,K} + \frac{\alpha}{K}\right)$
21	$\mu_k \sim \mathcal{N}(\dot{\mu}_k, \dot{\sigma}_k)$	**21**	$\mu_k \sim \mathcal{N}(\dot{\mu}_k, \dot{\sigma}_k)$
22		**22**	$(\pi_{k,1}, \ldots, \pi_{k,K}) \sim \text{Dirichlet}(\dot{\pi}_{k,1}, \ldots, \dot{\pi}_{k,K})$

Often, a preferred alternative to the collapsed Gibbs sampler for sequential models is the *forward-filtering backward-sampling* algorithm (FF-BS)(Scott, 2002). The FF-BS algortihm does not introduce coupling be-

tween the samples of the sequence of indicators $z_1, \ldots, z_T$, so the whole sequence can be sampled in one block, using efficient recursive updates (performing *dynamic optimization*[1]). For time series applications this can often improve mixing times dramatically (Scott, 2002). We briefly outline FF-BS in Algorithm **5.2**. The underlying recursion in FF-BS is a special case of a more general class of inference algorithms for graphical models Baum *et al.* (1970) which have not yet been fully explored.

Despite their widespread use, HMMs tend to suffer from several practical issues: MCMC methods for inference in HMMs (Scott, 2011) are computationally demanding and are slow to converge; furthermore the choice of K can be difficult. As with the case of GMMs, regularization heuristics such as AIC and BIC can be used, but we have discussed in Chapter 1 their pitfalls which also extend to sequential models. In the next sections we study methods for BNP modeling of sequential data which relax the assumption of fixed K in a statistically rigorous way.

5.3 Nonparametric Bayesian HMM

Consider the mixture distribution (5.1) that models each observation given state indicators at previous times in the sequence. In the BNP setting we would wish to relax the assumption of fixed K in (5.1) and model the probability of the data at any point in time with an infinite mixture. An obvious way to do this would be place a DP prior over the mixture weights $\pi_{z_{t-1},\cdot}$ and the component parameters θ in (5.1) (see Chapter 2). Assume that we draw a probability measure $G_{z_{t-1}} \sim \mathrm{DP}(N_0, G_0)$, we can then write $G_{z_{t-1}} = \sum_{k=1}^{\infty} \pi_{z_{t-1},k} \delta_{\theta_k}$ with G_0 being the conjugate prior over the component parameters of the HMM and z_{t-1} points to one of the represented states. In Chapter 2 we established how this notation implies that all component parameters $\theta_1, \ldots, \theta_K, \ldots$ are drawn i.i.d from the base measure H and the weights $\pi_{z_{t-1},1}, \ldots, \pi_{z_{t-1},K}, \ldots$ are generated using a stick-breaking construction. For different t the HMM implies that we need different measures G_{z_t} for each represented state to incorporate time dynamics into the model. However if we sample different measures G_k from a $\mathrm{DP}(N_0, G_0)$ the component parameters θ will be different for different G_k. This trivial setup would not allow for sharing of the state components across time and will end up producing too many different states and state parameters.

Now consider that G_0 is itself a probability measure drawn from a DP: $G_0 \sim \mathrm{DP}(M_0.H)$ with concentration parameter M_0 and base measure H which is the conjugate prior over the component parameters of the HMM. We can express the probability measures underlying this nonparametric HMM as:

$$\begin{aligned} G_0 &\sim \mathrm{DP}(M_0, H) \\ G_k &\sim \mathrm{DP}(N_0, G_0) \end{aligned} \tag{5.3}$$

where $k = 1, \ldots, K$ and K now denotes the number of represented states; N_0 and M_0 are *local* and *global* concentration parameters which can be thought as prior counts for the local and global DP respectively. This is exactly the setup we had in (4.1). There is a different measure G_k associated with each row of the transition matrix which implies different weights $\pi_{k,1}, \ldots, \pi_{k,K}, \pi_{k,K+1}$ in the transition matrix. However, the same time the component parameters θ are shared across $G_1, \ldots, G_K$ as their base measure G_0 is discrete.

This BNP extension of the HMM was named *HDP-HMM* in Teh *et al.* (2006) and the *infinite HMM* in Beal *et al.* (2002). Historically, the infinite extension of the HMM was first introduced in Beal *et al.* (2002), but later Teh *et al.* (2006) formalized the model in a fully Bayesian way with the use of HDPs. The infinite

[1] Dynamic optimization is an efficient method for solving a complex problem by breaking the problem down into a collection of simpler sub-problems and solving each of those sub-problems just once, storing their solutions. In inference methods it is typically implemented through recursive updates of the random variables in the model.

HMM from Beal *et al.* (2002) is equivalent to the HDP-HMM with CRF representation of the HDP where mixing parameters have been integrated out for both global and local DPs. Van Gael (2012) formally proved this relationship, but to avoid confusion in this thesis we will use both the names 'HDP-HMM' and 'infinite HMM' interchangeably. We will carefully specify any Rao-Blackwellization steps and any differences in the construction of the model where relevant.

5.3.1 Gibbs sampling methods for the HDP-HMM

Depending on the construction we use for the HDP for the transition mechanism in the HDP-HMM, we can derive different inference algorithms. In this section we review a Gibbs algorithm using the direct assignment representation of the HDP-HMM (Teh *et al.*, 2006) and the *beam sampling* algorithm (Van Gael *et al.*, 2008) which uses a complete, non-collapsed construction of the HDP-HMM. For completeness we also include a summary of the CRF-based Gibbs sampler for HDP-HMM which is given in Appendix I.

Direct assignment

The direct assignment representation of the HDP integrates over the local measures G_k underlying the transition matrix of the HMM. This means that each of the rows of the transition matrix π are Rao-Blackwellized and we refer explicitly to the state indicators z, the global mixture weights β and the component parameters θ. Here, β has the same meaning as in Chapter 4 where we express the base measure $G_0 = \sum_k^\infty \beta_k \delta_{\theta_k}$ and further combine the weights of all unrepresented components $\beta_{K+1}, \ldots, \beta_\infty$ into the term $\beta_{K+1} = \sum_{k=K+1}^\infty \beta_k$. As we showed in Chapter 4, the agglomerative property of the global DP then implies the following posterior for β:

$$(\beta_1, \ldots, \beta_K, \beta_{K+1}) \sim \text{Dirichlet}\,(M_1, \ldots, M_K, M_0) \tag{5.4}$$

where M_k can be interpreted as how many times the transition to state k has been drawn from the global DP.

The counts M_k are updated by sequentially sampling from the Polya urn scheme defining the probability of successive indicators:

$$p\,(z_{t+1} = k\,|z_j = p\,) \propto \begin{cases} N_{j,k} & \text{for an existing transition from } p \text{ to } k \\ N_0 \beta_k & \text{for a new transition from } p \text{ to } k \end{cases} \tag{5.5}$$

where, as in Section 4.3.3, we sample from (5.5) $N_{j,k}$ times, gradually increasing $N_{j,k}$ and we keep the count $M_{j,k}$ of how many times the transition from the second term has been sampled; we marginalize over all states that could have created a transition to k: $M_k = \sum_{j=1}^K M_{j,k}$. It is straightforward to see that if $N_{j,k} \neq 0$ then $M_{j,k} \neq 0$, because at least the first time a transition occurs from state j to state k, it has to be sampled from the global DP. As with Section 4.3.3 we can also sample $M_{j,k}$ using the expression in (4.16) from Chapter 4.

For every $t = 1, \ldots, T$ we update the state indicators using the posterior:

$$p\,(z_t = k\,|x, z_{-t}\,) \propto p\,(z_t = k\,|z_{-t}, \beta\,)\, p\,(x_t|z_t = k, \theta_k) \tag{5.6}$$

for each $k \in \{1, \ldots, K, K+1\}$ where z_{-t} denotes all indicators excluding the one for point t. The component likelihood term $p\,(x_t|z_t = k, \theta_k)$ in (5.6) is from the exponential family and is evaluated in exactly as in earlier models:

$$p\,(x_t|z_t = k, \theta_k) = \exp\left(\langle g\,(x_t)\,, \theta_k\rangle - \psi\,(\theta_k) - h\,(x_t)\right) \tag{5.7}$$

To compute $p(z_t = k \,|z_{-t}, \beta)$ from (5.6) we use the fact that given the global DP mixing weights β, $p(z_t = k \,|z_{-t}) = p(z_t = k \,|z_{t-1})\, p(z_{t+1} \,|z_t = k)$. This allows us to write out the state indicator probabilities:

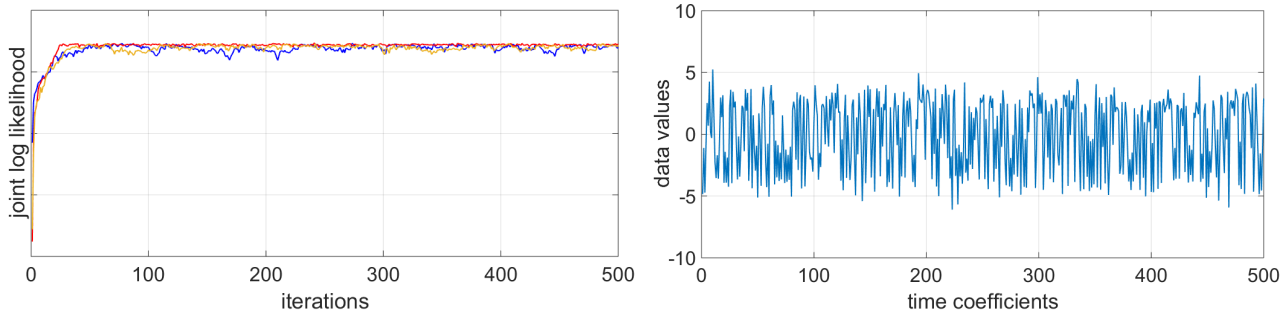
$$p(z_t = k \,|z_{-t}\beta) \propto \begin{cases} \left(N^{-t}_{z_{t-1},k} + N_0\beta_k\right) \frac{\left(N^{-t}_{k,z_{t+1}} + N_0\beta_{z_t+1}\right)}{N^{-t}_{k,\cdot} + N_0} & \text{for } k \leq K,\ z_{t-1} \neq k \\ \left(N^{-t}_{z_{t-1},k} + N_0\beta_k\right) \frac{\left(N^{-t}_{k,z_{t+1}} + 1 + N_0\beta_{z_t+1}\right)}{N^{-t}_{k,\cdot} + N_0 + 1} & \text{for } z_{t-1} = z_{t+1} = k \\ \left(N^{-t}_{z_{t-1},k} + N_0\beta_k\right) \frac{\left(N^{-t}_{k,z_{t+1}} + N_0\beta_{z_t+1}\right)}{N^{-t}_{k,\cdot} + N_0 + 1} & \text{for } z_{t-1} = k \neq z_{t+1} \\ N_0\beta_k\beta_{z_t+1} & \text{for } k = K+1 \end{cases} \tag{5.8}$$

where we have used the count $N_{k,\cdot} = \sum_{j=1}^{K} N_{k,j}$ to denote the total number of transitions from state k. The expression for the probability of the current state indicator z_t is obtained by computing the product of the CRPs placed on both transition to z_t and transition out of z_t. The second factor in this probability might need to be adapted in the cases when the previous state z_{t-1} was the same as the current one, $z_{t-1} = z_t$. This is because when $z_{t-1} = z_t = k$, the count $N_{k,z_{t+1}}$ must be updated accordingly to account for the transition that has occurred out of state k. When $z_{t+1} = z_{t-1} = k$ additional reinforcement is added and $N_{k,z_{t+1}}$ must be increased by 1, while when $z_{t+1} \neq z_{t-1} = k$ only the normalization is adapted.

As the component parameters θ and the mixing parameters β have not been integrated out, each time a new state is created, we need to sample a new θ_{K+1} and update the values of β accordingly. According to the stick-breaking construction of the global DP, whenever a new component is instantiated we need to update the value for β_{K+1} and create a new β_{K+2} using the stick-breaking process:

$$\begin{aligned} \nu \sim & \text{Beta}(1,\, M_0) \\ \beta_{K+2} = & \beta_{K+1}(1-\nu) \\ \beta_{K+1} = & \quad \beta_{K+1}\nu \end{aligned} \tag{5.9}$$

where β_{K+2} represents the updated weight of the unrepresented states in the top level DP.



(a) Joint Log Likelihood (b) raw data

Figure 5.2: The joint log likelihood of the direct assignment Gibbs sampler and the beam sampler (with uniform (in yellow) and Beta (in red) distributed auxiliary) for HDP-HMM applied to single dimensional Gaussian HMM data. On the same machine, using MATLAB implementation, the Gibbs sampler takes approximately 16 seconds compared to 9 and 4.5 seconds for the beam sampler with Beta and uniform auxiliary variables, respectively.

Beam sampling

The direct assignment representation above requires integrating out the local probability measures $G_1, \ldots, G_K$ from the complete HDP-HMM. The other well known CRF construction of the HDP requires integrating out both the probability measures $G_1, \ldots, G_K$ and the top level G_0. This means that both the direct assignment sampler and the CRF-based sampler for the HDP-HMM do not update the transition matrix π explicitly and we lose some of the conditional independence between the indicator variables. As discussed in Section 5.2 for sequential models, coupling between the indicator variables can lead to a lot less efficient sampling and prohibit us from exploring efficient recursive updates.

A common problem with BNP models such as the DPMM and the HDP-HMM is that the data likelihood is not tractable in its complete non-collapsed form as it supports an infinite number of components. To tackle this problem in the case of DPMMs Walker (2007) proposed a *sampling scheme* which introduces additional auxiliary variables to the model. The auxiliary variables by design do not change the probability of the data given the model (the marginal likelihood), but conditioned on the auxiliary variables the posterior distribution of the remaining model parameters are simplified. In the slice sampler described in Walker (2007), given a full draw of the auxiliary variables, the DPMM model parameters can be updated as it is a parametric Bayesian mixture model. As we re-sample the auxiliary variables, different parametric updates are effectively obtained for the variables of the model. At the same time the stationary distribution of the sampler from Walker (2007) converges asymptotically to the unbiased DPMM posterior.

Van Gael *et al.* (2008) used the same motivation to derive an auxiliary variable sampler for the HDP-HMM which was named the *beam sampler*. In BNP sequential models auxiliary variable samplers can be particular useful as they enable efficient recursive updates as with FF-BS for the parametric HMM. The beam sampler can be seen as an extension of FF-BS for inference in the nonparametric HDP-HMM[2].

The beam sampler from Van Gael *et al.* (2008) and the slice sampler from Walker (2007) both extend earlier work from Damlen *et al.* (1999) and Edwards & Sokal (1988) on Gibbs sampling in non-conjugate or less tractable probabilistic models. Edwards & Sokal (1988) and later Damlen *et al.* (1999) proposed various strategies to sample from hard to compute posterior distributions using auxiliary variables. Such methods borrow some ideas from *rejection sampling* (Gilks & Wild, 1992) techniques, but they are a lot easier to derive and usually do not inherit the notoriously slow convergence often associated with rejection sampling methods.

We now proceed with a formal description of the beam sampler. The method iteratively samples the auxiliary variables $u_1, \ldots, u_T$, state assignments $z_1, \ldots, z_T$, transition matrix π, the shared DP mixture weights $\beta_1, \ldots, \beta_K$ and component parameters $\theta_1, \ldots, \theta_K$. The distribution of the auxiliary variables is chosen by design such that the marginal likelihood of the model (the probability of the data given the model parameters) remains the same as for the original HDP-HMM. For example Van Gael *et al.* (2008) proposed using $u_t \sim \text{Uniform}\left(0, \pi_{z_{t-1}, z_t}\right)$ for $t = 1, \ldots, T$ as a robust and practical choice for distribution of u. Most of the values of $u_1, \ldots, u_T$ closer to 0 would lead to more expensive iterations but sometimes faster convergence and conversely larger u's imply cheaper iterations, but more iterations to convergence.

An alternative choice for the distribution of u is $u_t \sim \pi_{z_{t-1}, z_t} \text{Beta}\left(a, b\right)$ which introduces two new hyperparameters a and b controlling the distribution of the auxiliary variables (Van Gael, 2012). The beta distributed auxiliary variables will be a particularly useful model assumption to consider when it comes to efficient iterative MAP inference in the HDP-HMM. As the mode of the beta distribution is available in closed

[2]Another widely used extension of the FF-BS recursive sampler for HDP-HMM can be found in Fox *et al.* (2011), which is obtained by truncating the maximum number of states in the underlying HDP. Effectively, this truncation transforms the HDP-HMM into a parametric HMM where number of states is fixed to a large value.

form as long as $a > 1$ and $b > 1$, this version of the beam sampler will be a convenient starting point for our iterative MAP method for HDP-HMM which does not require any Rao-Blackewellization and can take advantage of efficient recursive updates (Section 5.4.2). The beam sampler iterates between the following updates:

- For each point t sample the auxiliary variables from $u_t \sim \text{Uniform}\left(0, \pi_{z_{t-1},z_t}\right)$, or alternatively $u_t \sim \pi_{z_{t-1},z_t}\text{Beta}\left(a,\, b\right)$.

- For each t, compute the posterior of the indicator z_t given data up to time t:

$$\begin{aligned} q_{t,k} = p\left(z_t = k \,\middle|\, (x_i)_{i=1}^{t}, (u_i)_{i=1}^{t}, \pi, \theta\right) &\propto p\left(x_t \,|\theta_k\right) \sum_{j=1}^{\infty} \mathbb{I}\left(u_t < \pi_{j,k}\right) p\left(z_{t-1} = j \,\middle|\, (x_i)_{i=1}^{t-1}, (u_i)_{i=1}^{t-1}, \pi, \theta\right), \\ &= p\left(x_t \,|\theta_k\right) \sum_{j:u_t<\pi_{j,k}} p\left(z_{t-1} = j \,\middle|\, (x_i)_{i=1}^{t-1}, (u_i)_{i=1}^{t-1}, \pi, \theta\right) \end{aligned} \tag{5.10}$$

  for all components $k = 1, \ldots, K, K+1$ and where we use $p\left(z_1 \,|x_1, u_1, \pi, \theta\right) \propto \sum_{k':u_1<\pi_{1,k'}}^{\infty} \pi_{1,k'} p\left(x_1 \,|\theta_k\right)$ for the first indicator. We have used the fact that the conditional probability of any u_t is $p\left(u_t \,|z_{t-1}, z_t, \pi\right) = \frac{\mathbb{I}\left(0<u_t<\pi_{z_{t-1},z_t}\right)}{\pi_{z_{t-1},z_t}}$ with $\mathbb{I}\left(\cdot\right)$ denoting the indicator function[3]. The sum in the probability in Equation (5.10) essentially defines a recursive relation between the indicators, i.e. the probability of the indicator for time point t is defined in terms of the probability of the indicator for time point $t-1$. Once we have computed all the probabilities $p\left(z_t = k \,\middle|\, (x_i)_{i=1}^{t}, (u_i)_{i=1}^{t}, \pi, \theta\right)$ for each t we proceed to sampling the whole trajectory z consisting of values for $z_1, \ldots, z_T$. We sample z_T from $p\left(z_T \,\middle|\, (x_i)_{i=1}^{T}, (u_i)_{i=1}^{T}, \pi, \theta\right)$ and perform a backward pass where we sample z_t given the sample for z_{t+1} using:

$$\begin{aligned} p\left(z_t = k \,\middle|\, z_{t+1}, (x_i)_{i=1}^{T}, (u_i)_{i=1}^{T}, \pi, \theta\right) &\propto q_{t,k} p\left(z_{t+1} \,|z_t = k, u_{t+1}, \pi\right), \\ &= \sum_{k'=1}^{\infty} \mathbb{I}\left(u_{t+1} < \pi_{k,k'}\right) \pi_{k,k'} q_{t,k} \\ &= \sum_{k':u_{t+1}<\pi_{k,k'}} \pi_{k,k'} q_{t,k} \end{aligned} \tag{5.11}$$

  where the first term $q_{t,k} = p\left(z_t = k \,\middle|\, (x_i)_{i=1}^{t}, (u_i)_{i=1}^{t}, \pi, \theta\right)$ has already been computed using (5.10).

- We sample the transition matrix from the conditional with respect to z and β:

$$\left(\pi_{k,1}, \ldots, \pi_{k,K}, \sum_{k'=K+1}^{\infty} \pi_{k,k'}\right) \sim \text{Dirichlet}\left(N_{k,1} + N_0\beta_1, \ldots, N_{k,K} + N_0\beta_K, N_0 \sum_{i=K+1}^{\infty} \beta_i\right) \tag{5.12}$$

- To update the top level mixing parameters β we first compute the counts $M_1, \ldots, M_K$ in exactly the same way we did in Section 5.3.1: either using the Polya urn scheme in (5.5) or sampling their values directly using (4.16). After we have updated the counts, $\beta_1, \ldots, \beta_K, \beta_{K+1}$ are sampled from the corresponding Dirichlet posterior:

$$\left(\beta_1, \ldots, \beta_K, \beta_{K+1}\right) \sim \text{Dirichlet}\left(M_1, \ldots, M_K, M_0\right) \tag{5.13}$$

[3]The indicator function, also sometimes known as the characteristic function indicates membership of an element to a particular set. The indicator function $\mathbb{I}\left(a < x < b\right)$ takes value of 1 whenever $x \in (a, b)$ and takes value of 0 whenever $x \notin (a, b)$. The indicator function is a generalization of the Kronecker delta we used earlier where for example we can express the Kronecker delta notation $\delta_{x,k}$ as $\mathbb{I}\left(k \leq x \leq k\right)$.

- The component parameters $\theta_1, \ldots, \theta_K$ are independent conditioned on z and x, i.e. we can update the values for all θ in parallel using the same updates as in previous sections.

Evaluation

We generate data from a Gaussian HMM in order to evaluate the performance of Gibbs and the beam samplers. The same data is also used in later sections to compare the performance of MCMC methods with related SVA and iterative MAP algorithms. We use a Gaussian HMM to generate univariate data with component variance of $\sigma = 0.5$. There are four fixed components with mean parameters centered at $\mu_1 = -4.0$, $\mu_2 = -1.0$, $\mu_3 = 2.0$ and $\mu_4 = 3.0$. The transition probabilities are equivalent for each state assuming 0.97 probability of self-transition and equal probability of transition to any of the remaining three states. In Figure 5.2 we plot the generated data; we also plot the joint likelihood of the beam sampler with both types of auxiliary variables we discussed and the joint likelihood of the direct assignment Gibbs sampler. The aim of this plot is to demonstrate the similar convergence behaviour of both Gibbs and beam samplers. Taking the segmentation of the samplers which is most likely (maximizes the joint posterior), we evaluate the NMI between estimated and true state indicators. The direct assignment Gibbs scores NMI of 0.93 and both beam samplers score NMI of 0.94. Using Matlab R2014b 64-bit on Windows 7 PC with i7-4770S CPU with up to 3.90GHz processor the Gibbs sampler took approximately 16 seconds to complete 500 iterations, the beam sampler with the beta distributed auxiliary variables took 9 seconds and the beam sampler with the uniform auxiliary variables took 4.5 seconds.

State persistence in HDP-HMM

In this section we discuss one serious limitation of the HDP-HMM which is that HDP-HMM inadequately models the temporal persistence of states (Fox *et al.*, 2011). On one side this is a general problem with HMMs, found in both classical parametric HMMs and non-parametric HMMs. However, in the non-parametric setting the problem is worsened because of the tendency for the DP to instantiate too many many components (Miller & Harrison, 2013). Therefore, in some sequential data problems the Bayesian bias towards simpler models is insufficient to prevent the HDP-HMM from giving high posterior probability to models with unrealistically rapid switching between states. In such problems we may want to consider some extensions of the HDP-HMM with a more flexible emission mechanism. For example, instead of modeling each state with a single exponential family distribution (as we did above) we can use more complex models such as a mixture of exponential family distributions, or an autoregressive model (Fox *et al.*, 2009).

We may also want to augment the HDP-HMM to include a parameter for self-transition bias. Fox *et al.* (2011) proposed a modified HDP and used it to construct a *sticky HDP-HMM* which places higher probability on self-transitions than the original HDP-HMM. Assuming the direct assignment construction the posterior update for the indicators from 5.8 changes for the case of the sticky HDP-HMM to:

$$p\left(z_t = k \,|z_{-t}, \varepsilon\right) \propto \begin{cases} \left(N^{-t}_{z_{t-1},k} + N_0\beta_k\right) \frac{\left(N^{-t}_{k,z_{t+1}} + N_0\beta_{z_t+1}\right)}{N^{-t}_{k,\cdot} + N_0 + \varepsilon} & \text{for } k \leq K,\ z_{t-1} \neq k,\ z_{t+1} \neq k \\ \left(N^{-t}_{z_{t-1},k} + N_0\beta_k + \varepsilon\right) \frac{\left(N^{-t}_{k,z_{t+1}} + N_0\beta_{z_t+1} + \varepsilon + 1\right)}{N^{-t}_{k,\cdot} + N_0 + \varepsilon + 1} & \text{for } z_{t-1} = z_{t+1} = k \\ \left(N^{-t}_{z_{t-1},k} + N_0\beta_k + \varepsilon\right) \frac{\left(N^{-t}_{k,z_{t+1}} + N_0\beta_{z_t+1}\right)}{N^{-t}_{k,\cdot} + N_0 + \varepsilon} & \text{for } z_{t-1} = k \neq z_{t+1} \\ \left(N^{-t}_{z_{t-1},k} + N_0\beta_k\right) \frac{\left(N_0\beta_{z_{t+1}} + N^{-t}_{k,z_{t+1}} + \varepsilon\right)}{N^{-t}_{k,\cdot} + N_0 + \varepsilon} & \text{for } z_{t-1} \neq k,\ z_{t+1} = k \\ N_0\beta_k\beta_{z_t+1} & \text{for } k = K+1 \end{cases} \tag{5.14}$$

where the new term ε now accounts for the newly incorporated self reinforcement effect; higher ε places higher probability on states being persistent across time. If the parameter ε is fixed, the updates for the other variables in a sticky HDP-HMM are nearly identical as for the original HDP-HMM.

The same self-reinforced probabilities over the state indicators was independently proposed in Beal *et al.* (2002), but Beal *et al.* (2002) introduced the self-transition parameter ε heuristically and did not formulate a fully Bayesian probabilistic model. Starting from a sticky HDP-HMM we can trivially extend the iterative MAP methods we develop in Section 5.4.2 to account for self-transitions. The beam sampler from Section 5.3.1 can be also modified to a state persistent HDP-HMM, by simply incorporating ε to put higher weight on the event of self-transitions.

5.4 Deterministic methods

MCMC methods for inference in the BNP extension of the HMM can often be prohibitively slow and this has motivated a lot of work on scaling up inference for the HDP-HMM. Most of the effort in this area has focused on deriving various VB inference algorithms. For example, Johnson & Willsky (2014) proposed a general framework for *stochastic variational inference* (SVI) for Bayesian sequential models in which stochastic gradient descent replaces the coordinate ascent as an optimization algorithm for a variational bound. As a result the data can be split into smaller batches and we can use a batch of data at a time for each update, leading to cheaper iterations. Johnson & Willsky (2014) has made a restrictive assumption that these batches of data are independent sets, where Foti *et al.* (2014) proposed a more general framework incorporating the dependence across the whole series of data. Both SVI schemes require knowledge of the size of the data a priori and further we need to store for each observation its variational distribution over the cluster assignments. This can be quite a restrictive requirement for many streaming applications especially where the memory of our computational hardware is constrained. Tank *et al.* (2015) proposed an SVI approach particularly for streaming applications, but focused on *normalized generalized Gamma processes* (NGGP) rather than particular time series models. Overall, streaming SVI methods involve a single sweep through a part of the data and can often lead to poor local solutions, as discussed for the case of DPMMs in Chapter 3. If we depart from the streaming setup and assume data can be revisited, Hughes *et al.* (2015b) proposed a novel VB bound which allows for processing data a batch at a time (as with (Johnson & Willsky, 2014; Foti *et al.*, 2014)), but avoiding zero variance point estimates at the top level DP. In addition, Hughes *et al.* (2015a) introduced a memoization step which keeps track of additional sufficient statistics of each batch resulting in a slight memory overhead, but often leading to significantly improved performance. For streaming applications the methods from Hughes *et al.* (2015a) and Foti *et al.* (2014) become comparable.

In contrast to exhaustive MCMC methods and VB methods that rely on factorization and truncation

assumptions, in this section we focus on simple SVA and iterative MAP methods for efficient inference in HDP-HMM.

5.4.1 SVA analysis for HDP-HMM

SVA-iHMM

Much as we extended the HDP to model sequential data, the SVA-HDP algorithm from Section 4.4.1 can be extended to an algorithm for fitting infinite HMMs to a time series. If we start from an HDP-HMM with exponential family states and we mirror the assumptions described in Section 4.4.1, we can obtain a deterministic algorithm which minimizes the following objective function:

$$\sum_{t:z_t=k} \sum_{k=1}^{K^{global}} D_\phi\left(x_t, \mu_k\right) + \lambda_1 \sum_{j=1}^{K^{global}} K_j^{local} + \lambda_2 K^{global} \tag{5.15}$$

with respect to state indicators z, expectation parameters $\mu_1, \ldots, \mu_{K^{global}}$, K^{global} and K_j^{local}; K_j^{local} denotes the number of states for which a transition from state j exists, and K^{global} denotes the total number of represented states. In this setup λ_2 penalizes the creation of new states where λ_1 controls how likely new transitions are between existing states. A simple iterative procedure that minimizes 5.15 can be obtained trivially by updating the steps of the SVA-HDP from 4.4.1 and we call this algorithm *SVA-iHMM*. As discussed in previous sections, this SVA-iHMM algorithm will not allow for standard model selection techniques in order to choose λ_1 and λ_2 and as with other SVA methods discussed in earlier chapters SVA-iHMM is purely geometric.

asymp-iHMM

Mirroring the derivation of K-means with reinforcement from Section 3.2.2 and the SVA algorithm with reinforcement for inference in DPMM (in Section 3.7.2) we can also derive a SVA method for inference in HDP-HMM. Roychowdhury *et al.* (2013) proposed such a method making all of the restrictive assumptions we already described in Section 3.2.2 and Section 3.7.2. The objective function which this method minimizes takes the following form:

$$\sum_{t:z_t=k} \sum_{k=1}^{K^{global}} D_\phi\left(x_t, \mu_k\right) - \lambda \sum_{t=2}^{T} \ln \frac{N_{z_{t-1},z_t}}{N_{z_{t-1},\cdot}} + \lambda_1 \sum_{p=1}^{K^{global}} K_p^{local} + \lambda_2 K^{global} \tag{5.16}$$

where T is the length of the data and λ is an additional threshold controlling the effect of the reinforcement of transitions. One easy way to minimize the objective in (5.16) is to mirror other SVA methods and derive an iterative procedure which looks like K-means or SVA-iHMM but has a few additional terms. However, such an approach for sequential data will be prone to falling into poor local optima. By contrast Roychowdhury *et al.* (2013) proposed to optimize (5.16) with a method which makes use of recursive updates, similar to FF-BS and beam sampling. Roychowdhury *et al.* (2013) called this method *asymp-iHMM* and unlike other SVA methods discussed in this thesis asymp-iHMM is not derived from any existing sampling methods. To a large extent, asymp-iHMM introduces additional heuristic assumptions to the standard SVA in order to allow for recursive inference. The method does not allow for model selection and loses the flexibility of the underlying HDP-HMM, but unlike SVA-iHMM it does not segment data purely on its geometry. We proceed with a summary of asymp-iHMM:

- For each each $t = 1, \ldots, T$ we complete the updates from 1. and 2. below:

1. We compute the distance matrix:

$$d_{j,k} = \begin{cases} \|x_t - \mu_k\|_2^2 - \lambda \ln\left(\frac{N_{j,k}}{N_{j,\cdot}}\right) & \text{using existing transition} \\ \|x_t - \mu_k\|_2^2 - \lambda \ln\left(\frac{1}{N_{j,\cdot}}\right) + \lambda U_j + \lambda_1 & \text{using a new transition} \\ \|x_t - \mu_k\|_2^2 - \lambda \ln\left(\frac{1}{T-1}\right) + \lambda_1 & \text{to a new state} \end{cases} \tag{5.17}$$

for all $j = 1, \ldots, K$ and $k = 1, \ldots, K$ where we have used $N_{j,\cdot} = \sum_{k=1}^{K} N_{j,k}$ to denote the total number of transitions out of state j; U_j is the upper bound of the possible change in a transition probability from state j that could incur by introducing a new transition to existing state. For $k' \in \{1, \ldots, K\}$ we have:

$$U_{j,k'} = (N_{j,k'} - 1)\left(\ln \frac{N_{j,k'}}{N_{j,\cdot}} - \ln \frac{N_{j,k'} - 1}{N_{j,\cdot}}\right) \tag{5.18}$$

and we use the minimum $U_{j,k'}$ for the upper bound, $U_j = \min_{k'} U_{j,k'}$. See (Roychowdhury *et al.*, 2013) for proof.

2. For all $k \in \{1, \ldots, K\}$ we compute the minimum sum of distances needed to reach state k:

$$q_{t,k} = \min_{1 \le j \le K} [q_{t-1,j} + d_{j,k}] \tag{5.19}$$

We also compute the cheapest transition $d_{min} = \min_{j,k} d_{j,k}$ and if $d_{min} > \lambda_1 + \lambda_2$ we create a new state. This involves computing:

$$q_{t,K+1} = \min_{1 \le j \le K} q_{t-1,j} + \lambda_1 + \lambda_2 \tag{5.20}$$

and increasing $K = K + 1$.

- After a full sweep through the observations is done iterating between Step 1. and Step 2. above for each time point, we again sweep through $t = 1, \ldots, T$ and update the indicators:

$$z_t = \arg\min_k q_{t,k} \tag{5.21}$$

- After all indicators z are updated, we proceed with the update step for the state centroids. For each $k = 1, \ldots, K$ we update the state centroids $\mu_k = \bar{x}_k$ with the sample mean of observations assigned to state k.

Roychowdhury *et al.* (2013) also proposed an additional step to the algorithm which considers assigning whole groups of observations to a new state. Similar extra steps aiming at more efficient inference in sequential models have also been proposed in Hughes *et al.* (2012) and Hughes *et al.* (2015b). The additional step from Roychowdhury *et al.* (2013) suggests we sweep through all pairs of states (j, k) such that transitions between state j and state k exists. Then we evaluate how much the objective function in 5.16 would change if all transitions from j to k are replaced with transitions from state j to a new state. Based on the effect in the objective function we consider re-assigning a group of the observations in state k to a new state. A detailed description of that extra move can be found in the Appendices of Roychowdhury *et al.* (2013).

Evaluation The asymp-iHMM was tested on the same data from Figure 5.2 (synthetic Gaussian HMM data). It scored an average NMI of 0.54 (using 25 restarts) with 20 iterations to convergence and we plot a

visual reconstruction produced by asymp-iHMM in Figure 5.3). The method is rather sensitive to the order in which it processes the time series and to maximize performance we often need to start from the middle of the time series. This can be a serious drawback, especially when considering online learning applications (see later, Chapter 6). From 25 restarts of the order in which we process the synthetic set, the best clustering with asymp-iHMM was NMI of 0.76 and most of the other clusterings scored NMI between 0.50 and 0.60. This performance was only reached using the additional "block re-assignment" heuristic discussed above. The MAP methods we propose in the next section have not yet been extended to include a *split and merge* step and this is likely to further improve performance. Significant drawbacks of asymp-iHMM are that like other SVA methods it implicitly assumes that the time series is spherical and it does not allow for standard model selection techniques in an unsupervised way. Some heuristics are suggested in the appendices of Roychowdhury *et al.* (2013), but they often assume we are approximately aware of the true number of states in the data. Roychowdhury *et al.* (2013) proposed a way of doing prediction on unseen data using asymp-iHMM, but this is done for one point at a time.

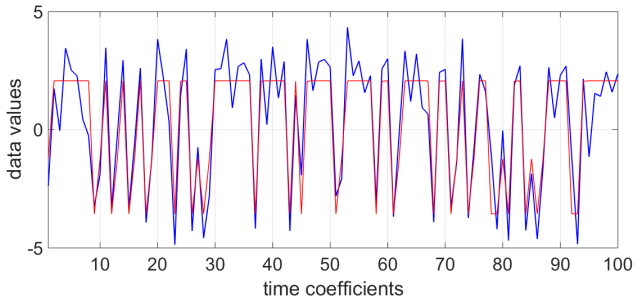


Figure 5.3: Data reconstruction using the inferred state indicators and state means. The raw data is plotted in blue. The asymp-iHMM method is used to learn state indicators and the centroids of points in each state. The red line replaces each data point with its corresponding centroids as inferred. The data is generated from a single dimensional Gaussian HMM.

5.4.2 Iterative MAP inference for iHMM

Based on the different constructions of the underlying HDP, we can also derive different iterative MAP algorithms for inference in the HDP-HMM. Using the CRF construction, we can derive a MAP method closely related to the CRF-based Gibbs sampler for HDP-HMM (Appendix I). However, this method is a simple adaptation of Algorithm 4.3 to the case of HDP-HMM where the number of sets J is replaced by the number of represented states K. We point out that while the number of sets J in the HDPs from Chapter 4 was fixed in advance, K changes throughout the iterations of HDP-HMM.

In this section we will derive MAP methods from the direct assignment construction (Section 5.3.1) of the HDP-HMM and from the auxiliary variable representation (Section 5.3.1).

MAP-iHMM

The MAP-iHMM algorithm (Raykov *et al.*, 2016b) we present here is derived using the direct assignment representation of the HDP-HMM, therefore it involves iterative updates of the mixing parameters β, the counts of the top level DP $M_1, \ldots, M_K$ and the state indicators z.

- The updates for $\beta_1, \ldots, \beta_K, \beta_{K+1}$ can be obtained by taking the mode of their Dirichlet distributed posterior (from (5.4)). As long as the top level concentration parameter (prior count) $M_0 > 1$ this

modal update is available in closed form giving the updates:

$$\begin{aligned} \beta_k &= \frac{M_k - 1}{\sum_{j=1}^{K} M_j + M_0 - K} && \text{for an existing state } k = 1, \ldots, K \\ \beta_{K+1} &= \frac{M_0 - 1}{\sum_{j=1}^{K} M_j + M_0 - K} && \text{for a new state} \end{aligned} \tag{5.22}$$

For the later steps, we can simplify the expressions if we replace β with $\hat{\beta}$ where $\hat{\beta}_k = M_k - 1$ for $k \in \{1, \ldots, K\}$ and $\hat{\beta}_{K+1} = M_0 - 1$. This is because we can drop the normalization term, as later, when optimizing the indicators, they will be independent of the state k.

Analysis of the assumption $M_0 > 1$:

The mode of the Dirichlet distribution is available in convenient closed form only when all of its parameters are larger than 1. In the posterior of β those parameters are the counts $M_1, \ldots, M_K, M_0$. It is natural to assume that each of the counts for any represented state will be at least 1 and if it is exactly 1 no reinforcement effect for that state is contributed from the global DP (because this is a spurious state with a single point in it). The assumption for $M_0 > 1$ in a probabilistic treatment of the HDP-HMM is somewhat of a limitation. This is because for certain problems small values of M_0 imply the correct posterior of interest. For example, in topic modeling applications and natural language processing, we typically fix the DP concentration parameter at very small values (Gal & Ghahramani, 2014). However, note that the concentration parameter in all iterative MAP methods plays a different role, the role of a prior count. Really small values of the concentration parameters of the underlying DPs might be useful to model really slowly changing dynamics if we use exhaustive MCMC methods for many thousands of iterations as Gal & Ghahramani (2014) showed, but using iterative MAP inference for such problems would just result in a poor local fit, independent of whether we fix the concentration parameter to be smaller or larger than 1.

- To update the counts $M_1, \ldots, M_K$ we take one of the following two approaches: we can numerically maximize the posterior of the counts from (4.16); we can use the Polya urn scheme (defined in (5.5)). To maximize the posterior of $M_1, \ldots, M_K$, we evaluate (4.16) substituting every value of $m \in \{1, \ldots, N_{j,k}\}$. We choose the value of m which maximizes the expression in (4.16) and this is our MAP value for $M_{j,k}$: $M_{j,k}^{MAP} = \arg\max p\left(M_{j,k} \,|z, \beta, N_{j,k}\right)$. As defined earlier, we use the notation $M_k = \sum_{j=k}^{K} M_{j,k}$ and that $M_{j,k}$ is always in the range between 1 and $N_{j,k}$. However, the posterior in (4.16) is hard to evaluate for more than a few thousand points in the time series so it is far more practical to use the Polya urn scheme from (5.5) to evaluate the counts M. This means mirroring the estimation step for M from Section 5.3.1. The stochastic nature of this update will result in minor fluctuations in the joint likelihood of MAP-iHMM after convergence, however they have very little effect on the MAP solution of MAP-iHMM and the method is trivially stopped (see Figure (5.4)).

- For each $t = 1, \ldots, T$ we compute the negative log probability of the state indicators (using (5.8)) for each existing state k and for a new state $K + 1$:

$$\begin{aligned} q_{t,k} = &\quad -\ln p\left(x_t \,|z_t = k, \theta_k\right) - \ln p\left(z_t = k \,|z_{-t}, \beta\right) \\ q_{t,K+1} = &\quad -\ln p\left(x_t \,|\tau_0, \eta_0\right) - \ln\left(N_0 \beta_{K+1} \beta_{z_t+1}\right) \end{aligned} \tag{5.23}$$

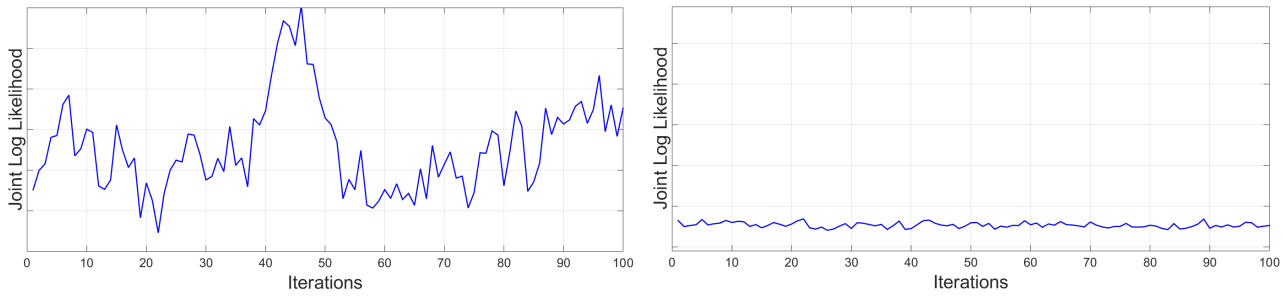
where again and without losing generality, we can omit the terms independent of k. The probabilities $p\left(x_t \,|z_t = k, \theta_k\right)$, $p\left(x_t \,|\tau_0, \eta_0\right)$ and $p\left(z_t = k \,|z_{-t}, \beta\right)$ are evaluated in the same way as in Section 5.3.1. For each t, we compute the $K + 1$-dimensional vector q_t and select the state number according to:

$$z_t = \underset{k \in \{1, \ldots, K, K+1\}}{\arg\min} q_{t,k} \tag{5.24}$$

- If a new state $K+1$ is chosen we need to instantiate a new state parameter θ and update β accordingly. We maximize over the posterior update of β defined in (5.9) and we choose new state parameters maximizing the posterior distribution of θ with a single observation. The update of β implies that when the new state $K+1$ is chosen, we create $\beta_{K+2} = \beta_{K+1}$ and update the older value of $\beta_{K+1} = 0$.

- The state parameters $\theta_1, \ldots, \theta_K$ are updated in the same way as in MAP-DPMM and MAP-HDP, using the mode of their corresponding posterior.

MAP-iHMM converges to a fixed point solution if we use numerical optimization for the update of the counts $M_1, \ldots, M_K$. Often it is more practical to update $M_1, \ldots, M_K$ stochastically as this solution is more memory efficient and enables us to process more data. In that case MAP-iHMM does not converge completely and we observe minor flucations in the joint likelihood even after convergence. Practically, the algorithm still falls into a local optima and we can still stop the method in a straightforward way once fluctuations in the objective function (the joint likelihood) fall bellow a certain threshold (see Figure 5.4). This convergence is easy to asses as the joint likelihood of MAP-iHMM is dominated by updates of z, β and θ and they are deterministic.



(a) MCMC (b) iterative MAP

Figure 5.4: Joint log likelihood at convergence for MCMC (Beam sampler) and iterative MAP (with stochastic Polya urn step). Convergence for the MCMC is reached after approximately 300 iterations compared to approximately 10 iterations for the MAP scheme.

Dynamic MAP-iHMM

Integrating over the transition matrix π and the mixing parameters β introduces couplings between the indicator variables in an HDP-HMM and prohibits us from using efficient recursion (like those used in FF-BS for parametric HMMs) to update the posterior over the indicators. The beam sampler from Section 5.3.1 provided a solution for this in the framework of dynamic MCMC inference algorithms for the HDP-HMM. In this section we use the beam sampler as a starting point to derive a faster and more efficient dynamic deterministic algorithm for inference in HDP-HMMs: the *dynamic MAP-iHMM*.

Assume the setup from Section 5.3.1 where we refer to the complete representation of the HDP-HMM with random quantities z, π, β, and θ and in addition we introduce auxiliary variables u. In order to obtain a closed-form MAP step for the update of u, we assume the following distribution of the auxiliary variables $u_t \sim \pi_{z_{t-1},z_t}\text{Beta}\,(a,\, b)$. The dynamic MAP-iHMM will iterate between modal updates for z, π, β, θ and u where the method will also make the same assumptions that MAP-iHMM does for more efficient updates of $\beta_1, \ldots, \beta_K, \beta_{K+1}$ and of the counts M. We list each of those updates:

- For each point t, we update $u_t = \pi_{z_{t-1},z_t}\frac{a-1}{a+b-2}$ using the mode of the beta distribution.

- For each point t we first compute the quantities:

$$q_{t,k} = p\left(x_t \left|\theta_k\right.\right) \sum_{j=1}^{\infty} \mathbb{I}\left(u_t < \pi_{j,k}\right) \frac{u_t^{a-1}\left(1-u_t\right)^{b-1}}{\mathrm{B}\left(a,\, b\right)} p\left(z_{t-1} \left| \left(x_i\right)_{i=1}^{t}, \left(u_i\right)_{i=1}^{t} \right.\right) \tag{5.25}$$

  for all $k = 1, \ldots, K, K+1$ where $\mathrm{B}\left(a,\, b\right)$ denotes the *beta function* with $\mathrm{B}\left(a,\, b\right) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$; θ_{K+1} is estimated using the prior hyperparameters. Once the $T \times (K+1)$ table q has been updated, we proceed to updating the whole sequence $z_1, \ldots, z_T$. The last indicator z_T is updated using $z_T = \arg\max_{k \in \{1,\ldots,K+1\}} q_{T,k}$; the remaining z_t (for $t = 1, \ldots, T-1$) are updated consecutively through a backward pass using:

$$\begin{aligned} z_t &= \underset{k \in \{1,\ldots,K+1\}}{\arg\max}\; q_{t,k} p\left(z_{t+1} \left| z_t = k, u_{t+1}, \pi\right.\right) \\ &= \underset{k \in \{1,\ldots,K+1\}}{\arg\max} \sum_{k' : u_{t+1} < \pi_{k,k'}} \pi_{k,k'} q_{t,k} \end{aligned} \tag{5.26}$$

  where $q_{t,k}$ has already been computed using (5.25). Note that the update for z_t is recursively defined through the values chosen for z_{t+1}.

- If a new state $K+1$ is chosen we need to instantiate a new state parameter θ and update β as for MAP-iHMM above.

- For each $k \in \{1, \ldots, K\}$ we update the rows of the transition matrix using:

$$\pi_{k,k'} = \begin{cases} \frac{N_{k,k'} + N_0\beta_{k'} - 1}{\sum_{j=1}^{K} N_{k,j} + N_0\beta_j + N_0\beta_{K+1} - K - 1} & \text{for represented component } k' \\ \frac{N_0\beta_{K+1} - 1}{\sum_{j=1}^{K} N_{k,j} + N_0\beta_j + N_0\beta_{K+1} - K - 1} & \text{for new } k' = K+1 \end{cases} \tag{5.27}$$

  for $N_0\beta_{K+1} - 1 > 1$ and if $N_0\beta_{K+1} - 1 < 1$ update $\pi_{k,K+1} = 0$; $k' = 1, \ldots, K$ denotes the represented components and the final column of the transition matrix reflects the transitions to a new state.

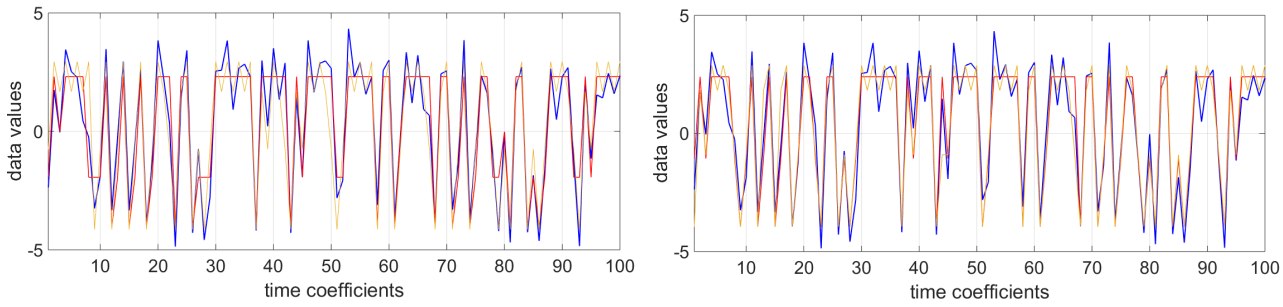
- To update the top level mixing parameters β, we first compute the counts $M_1, \ldots, M_K$ as for MAP-iHMM. Then we update the global mixing parameters β for each $k \in \{1, \ldots, K+1\}$:

$$\beta_k = \begin{cases} \frac{M_k - 1}{\sum_{j=1}^{K} M_j + M_0 - K} & \text{for existing } k \\ \frac{M_0 - 1}{\sum_{j=1}^{K} M_j + M_0 - K} & \text{for new } k = K+1 \end{cases} \tag{5.28}$$

- The state parameters $\theta_1, \ldots, \theta_K$ are updated using the mode of their corresponding posterior.

Evaluation In Table 5.1 we compare the performance of MAP-iHMM and dynamic MAP-iHMM using NMI. The NMI of the iterative MAP methods is contrasted to the NMI obtained by the corresponding MCMC method where we use the most likely draw from the corresponding chain. After 25 restarts of MAP-iHMM and dynamic MAP-iHMM the best solution was obtained using MAP-iHMM which scores an NMI value of 0.81. By contrast the best solution obtained using dynamic MAP-iHMM scores NMI of 0.66. However, dynamic MAP-iHMM showed to be more robust to initialization with NMI across the 25 restarts varying in the range of 0.56–0.66 with mean NMI of 0.61; MAP-iHMM often scores a lot worse with NMI in the range of 0.43–0.81 and mean NMI of 0.68.

Both MAP methods are significantly faster than the MCMC alternatives. Between the two MAP methods dynamic MAP-iHMM is faster with an average performance time of 0.08 seconds compared to average



(a) Dynamic MAP-iHMM (b) MAP-iHMM

Figure 5.5: Data reconstruction using the inferred state indicators and state means. The raw data is plotted in blue. MAP methods are used to learn state indicators and the centroids of points in each state. The red line replaces each data point with its corresponding centroids as inferred. For comparison the same re-construction has been applied using the corresponding MCMC methods to learn states and centroids.

Table 5.1: Infinite HMM estimation performance measured using NMI obtained using iterative MAP and MCMC methods applied to synthetic Gaussian HMM data. The NMI is measured between the true state indicators from the generating model and the indicators estimated by the different inference techniques. For the MAP methods we display the score of the best segmentation produced out of 25 restarts. For MCMC we display the NMI scored using the most likely draw from the joint posterior distribution from 500 iterations. In the brackets we report execution time which for iterative MAP, is the time of the single best scoring run ignoring restarts.

Algorithm	NMI score and run time
MAP-iHMM	0.81 (0.26 seconds)
Dynamic MAP-iHMM	0.66 (0.06 seconds)
Direct assigment Gibbs sampler	0.93 (16 seconds)
Beam sampler	0.94 (9 seconds)

execution time for MAP-iHMM of 0.27 seconds. The reported times in Table 5.1 are for the particular draw that scored the highest NMI. The reported run times for MAP and MCMC inference were obtained using Matlab R2014b 64-bit, i7-4770S CPU with up to 3.90GHz processor, Windows 7 PC.

Visual interpretation of the reconstruction provided by both MAP methods compared to their stochastic counterparts is shown in Figure 5.5. We have used 1 dimensional data to allow for intuitive visual interpretation of the methods, however as the dimensionality and the size of the data increases the difference in convergence between MCMC methods and MAP methods becomes a lot more significant.

5.5 Applications

5.5.1 Genomic hybridization and DNA copy number variation

In this study we consider the data assembled from Snijders *et al.* (2001) for measurement of DNA copy number across the human genome. The data comprises of 2316 *bacterial artificial chromosome* (BAC) clones for measures of DNA copy-number across the human genome. Genomic hybridization (segmentation analysis of the BAC clones) provides a means to quantitatively measure the copy-number aberrations and map them directly onto the genomic sequence. Arrays comprised of BAC clones provide reliable copy-number measurements on individual clones and this makes them potentially useful for clinical applications in medical

genetics and cancer.

We examine the potential of the MAP-iHMM algorithm for the problem of genomic hybridization. A common assumption to make is that each of the copy-number regimes are constant and there is some independent and stationary noise, also there are known to be a few outliers. We are interested in detecting a few, large jumps between different copy-number regimes. In practice different regimes of the copy-number indicate DNA samples from different sources for example differences between the chromosomal complements of solid tumor and normal tissue.

In Figure 5.6 we plot a reconstruction of the copy-number regimes using MAP-iHMM. We have used a Gaussian model where in red we plot the mean of the state with which each observation is associated. By varying N_0 and M_0 we can model at different granularity the changes in the copy-numbers. To get an idea of how MAP-iHMM compares to direct assignment Gibbs sampling for this problem we measure NMI between the MAP-iHMM solution from Figure 5.6 and the best solution obtained with Gibbs[4]. MAP-iHMM scored NMI of 0.58 after 24 iterations, where the Gibbs sampler was ran for 2500 iterations in total with the best solution obtained after 1256 iterations. Despite this, the MAP-iHMM solution departs substantially from the Gibbs solution, for the problem of genomic hybridization it is the more practical one. Due to the issues related to state persistence discussed in Section 5.3.1, the HDP-HMM suggests much too complicated dynamics which is not an accurate representation of the copy-number regimes. MAP-iHMM does not fit the HDP-HMM exactly and captures only the larger states and the most likely transitions. Therefore in this problem MAP-iHMM actually leads to a more practical reconstruction. For a more in depth study of the data set and visual benchmarks of the copy-number regimes, we refer the reader to (Little & Jones, 2011).

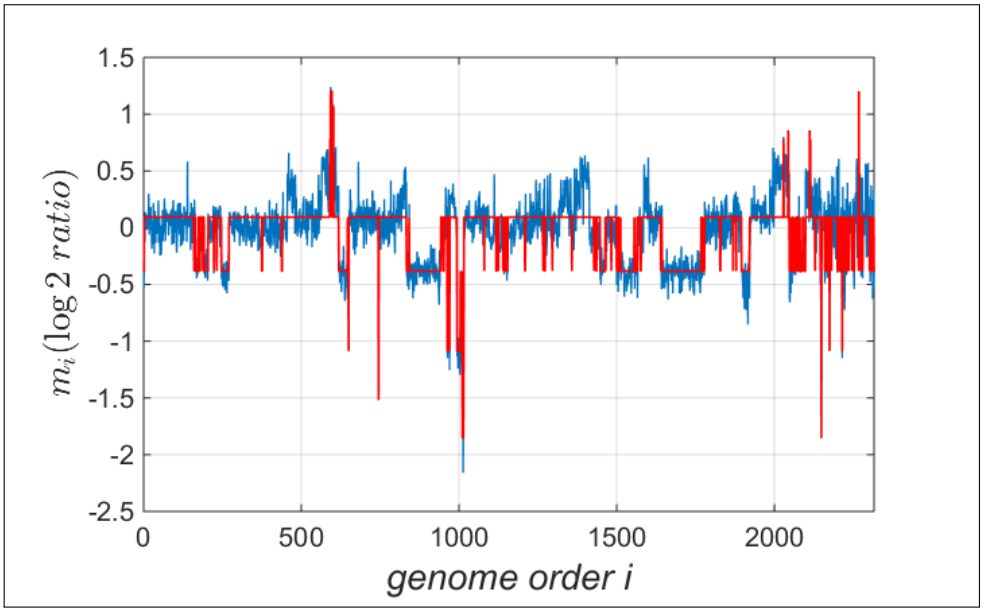


Figure 5.6: 2316 BAC clones (blue) for measurement of DNA copy number variation across the human genome. The red line is the reconstruction obtained using the MAP-iHMM algorithm.

[4]The best solution provided by the Gibbs sampler is the one that maximizes the empirical posterior, or the joint likelihood of the model.

5.5.2 Behaviour extraction from accelerometer data

In this section we demonstrate the potential of MAP-iHMM as an *activity recognition* tool that uses smartphone accelerometer data. The Oxford Parkinson's Disease Centre (OPDC), University of Oxford, UK have deployed an Android smartphone app that records data from a variety of smartphone sensors without any user interaction, i.e. the user puts the smartphone in their pocket and goes about their day-to-day activity while data from the sensors are passively recorded.

The mobile app consists of 5 "active tests" which require the user to carry out a set of instructions aimed at detecting characteristic Parkinson's disease (PD) symptoms such as Parkinsonian gait, tremor and postural instability. Here we focus on data from the gait test. During the gait test, users press a "Start" button on the screen and put the smartphone in their pocket, once the smartphone vibrates, users must walk in a straight line for 20 yards, turn 180 degrees and then walk back. The buzzer goes off again after 30 seconds to indicate the end of the test. The phone captures raw 3-axis accelerometer time series with sampling rate of 120Hz, see Figure 5.8(a).

Using the same mobile application, but deployed in a clinically-controlled setting Arora *et al.* (2014) demonstrated that feature extraction on the collected data, followed by a random forest classifier, is capable of distinguishing PD patients from healthy controls with an average sensitivity of 98.5% and an average specificity of 97.5%. This data however, was collected from a smaller number of participants and in a controlled environment. Once participants start doing the gait test in their home, outside, or under uncontrolled circumstances, the number of biasing behaviours that can affect the gait test accelerometer input can grow "infinitely" (or it is reasonable to assume so). Therefore, to maximize the performance of the approach in Arora *et al.* (2014) or any further classifications methods for diagnosis of PD, we wish to segment unwanted behaviors from the gait data. That is, to achieve higher accuracy with existing classifiers and understand better how to extract meaningful features, the data analysis needs to focus on the accelerometer data mainly from the gait (walking behaviour) ignoring any other behaviours or *orientation changes*[5] in the device. Manually segmenting the relevant parts of the data signal is possible, but costly and highly infeasible considering the large amounts of data generated by a single subject. For example, a single study can involve analyzing two or three gait tests per day, from 1000 individuals, for the duration of 60 days: generating 3-dimensional time series of $\sim 648,000,000$ observations.

We propose using the efficient and simple MAP-iHMM as a completely unsupervised pre-processing tool to segment the accelerometer data associated with gait from the rest of the data. Each state of the HDP-HMM is modeled with a Gaussian distribution with unknown mean and precision therefore assuming NW prior H over the state parameters. In Figure 5.8(b) we plot the segmentation suggested by the MAP-iHMM where we have tuned the model in an unsupervised way (see Appendix E). The red points are clustered together and they mostly match gait data, using hand labels of a specialist for comparison. The exception to this is the first 1 second of the signal clustered as walking, which in fact is just a vibration of the phone due to a buzzer indicating the start of each test. In blue the MAP-iHMM has clustered with high accuracy some of the unwanted behaviour such as long stops, the phone being put on a flat surface (the steady segments) and orientation changes (the re-ordering in the x-, y- and z-axis). This means that using MAP-iHMM we can extract raw data describing gait, and filter out data that is unwanted (nearly 100% of the blue clustered data). By removing possibly an infinite number of biasing behaviours, we significantly simplify the task of classification algorithms (such as deep neural networks, random forests etc.) to detect PD symptoms from

[5]By orientation changes we mean rotation of the device with respect to the Earth's gravitational field. When analyzing smartphone data, the quality of the conclusions we make about the processed data often depend upon being able to adequately account for orientation changes.

gait tests performed on a smartphone device.

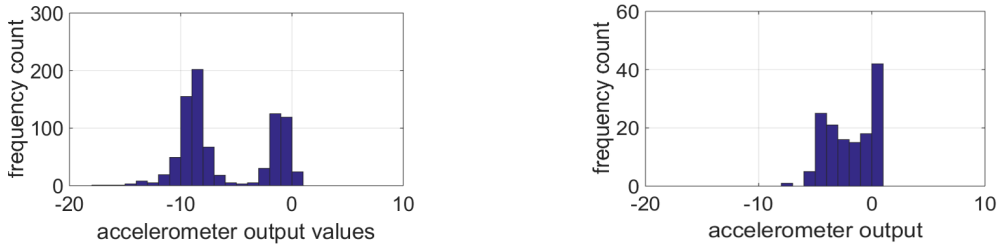
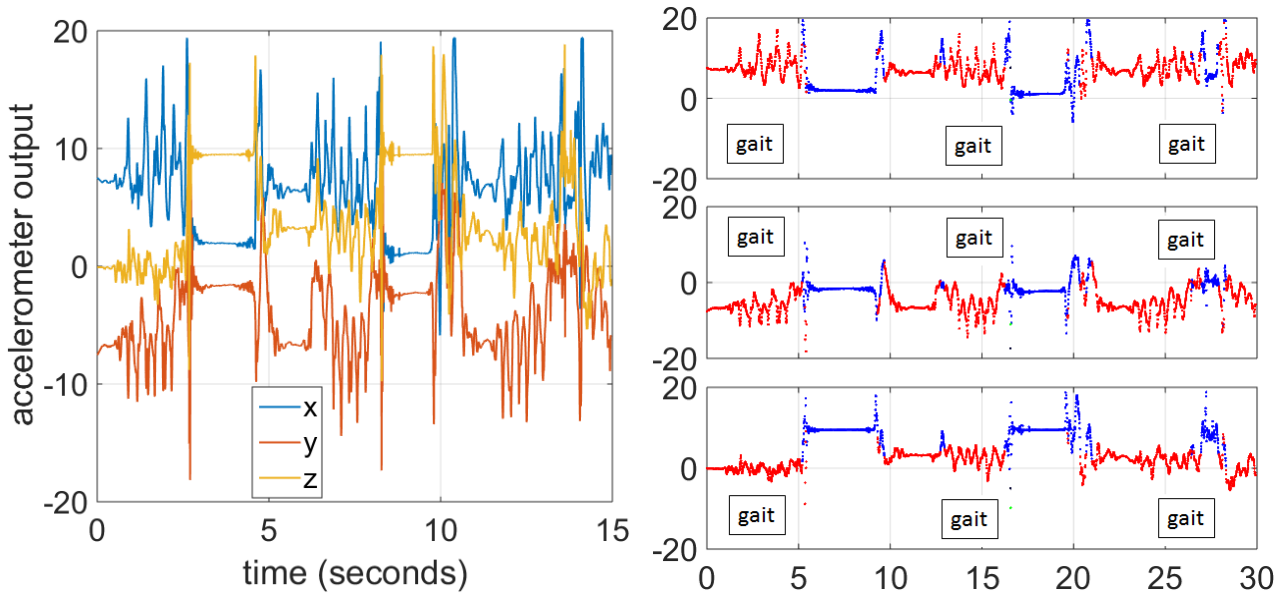


Figure 5.7: Distribution of accelerometer output from the gait for different tests.

Despite the great potential of the MAP-iHMM, we have also identified some challenges for it which mainly result from the model assumptions inherent in the HDP-HMM. MAP-iHMM with simple Gaussian states often fails to: (a) segment out the buzzer into a different state (or states); (b) handle orientations changes; (c) segment correctly walking patterns with very different dynamics. One of the major reasons for this is that the data describing gait rarely follows unimodal Gaussian distribution (see Figure 5.7). Therefore, if we tune the hyperparameters of the HDP-HMM with Gaussian states in such a way that we segment the buzzer output, the model suggests breaking down the gait data into multiple different states. A rigorous treatment to these challenges would be to extending the HDP-HMM to having a multimodal emission mechanism (e.g. DP mixture of Gaussians) and then incorporating iterative MAP for inference.



(a) Raw accelerometer output

(b) Clustered output

Figure 5.8: 3-axis accelerometer output from a 30 seconds walking test for detection and diagnosis of Parkinson's disease. The segmentation of the data allows us to automatically remove most of the unwanted user's behaviours and focus only on the walking data in the following feature extraction stage. In red is the actual part of the data we will be interested at.

5.6 Discussion

In this chapter we extend the iterative MAP approach for inference in sequential BNP models. In keeping with earlier chapters we contrasted our MAP methods to SVA algorithms and compared both with related sampling techniques on synthetically generated data. In this chapter we build on our previous discussion about the effect that Rao-Blackwellization can have for inference in different BNP models. Where in Chapter 3 we demonstrated mainly positive effects of integrating over component parameters in mixture models, here we advocate the opposite for sequential problems. Despite the theoretical arguments in favor of collapsed and non-collapsed model representation, in practice both models (and the related inference methods) have their advantages and we see that MAP-iHMM can outperform dynamic MAP-iHMM.

Although, we observe a drop in accuracy of iterative MAP for fitting sequential models, this is to be expected considering the deeper hierarchy of the underlying model. The solution usually obtained with MAP-iHMM and dynamic MAP-iHMM is unable to capture states and transitions which have small support under the HDP posterior. However, for long time series the alternative unbiased sampling methods converge only theoretically to a global solution.

We demonstrated example applications of iterative MAP for detecting DNA copy-number regimes and as a behaviour segmentation tool for quality control of accelerometer data. The simplicity, scalability and flexibility of model-based algorithms such as MAP-iHMM and dynamic MAP-iHMM make them extremely valuable in embedded applications where an initial structuring decision is to be made directly on a data collection device (like a smartphone). In the next chapter we introduce a novel sensing problem, for which we reach a sensible modeling solution exactly through the scalability of MAP-iHMM.

Chapter 6

Occupancy estimation using nonparametric HMMs

6.1 Introduction

The motivation for using greedy deterministic methods such as the ones discussed above is that there are many applications that can benefit from BNP models where computational resources are at premium and intensive MCMC inference limits the use of such models; for example if we want to model data directly onto some embedded hardware collecting it. In this chapter we study one problem like that and demonstrate how an application can really benefit from the insights that BNP models such as the infinite HMM can provide. At the same time for the application it is essential that a lot of the learning and modeling is performed on a microcontroller board with severely limited specifications, so most of the current state of the art inference methods are not feasible.

The problem we study is whether we can design a low cost system that can estimate the number of people in some monitored environment using only data from a single analogue *passive infrared* (PIR) sensor. We present some motivation and background of this problem of *occupancy estimation* in Section 6.1.1. In Section 6.2 we explain the experimental setup and conditions under which the data for our study has been collected where Section 6.3 summarizes a simple scalable approach to the problem which relies on basic feature extraction stage followed by regression stage. Then in Section 6.4 we show how the infinite HMM can be used to provide more meaningful solution and we quantify the benefits of that through rigorous evaluation in Section 6.6. In Section 6.7 we study the practical feasibility of the suggested system testing different inference methods and techniques for the infinite HMM.

6.1.1 Motivation

In contrast to existing occupancy estimation systems, we investigate the potential of using a single low-cost PIR sensor for counting the number of people inside of its *field of view*[1] and propose a novel system that relies on a single sensor to monitor a room. We extract motion patterns from the raw sensor data with an iHMM and use those patterns to infer the number of occupants using basic statistical regression methods. This system is well-suited to the adaptive setting on active deployment whereby the iHMM readily finds new motion patterns in the signal as new data arrives.

[1] The field of view of a sensor is the area of space directly seen from the sensor within its range.

We demonstrate the system configured to estimate an occupancy count for various time windows ranging from 30 seconds to 20 minutes. The result of these tests show that this approach can accurately estimate room occupancy count to within ± 1 for time windows of less than 2 minutes. We also explore the challenges imposed by using a single PIR sensor in terms of the monitored room size, maximum number of distinguishable occupants, and the restrictions imposed by the sensor' s range and view angle.

6.1.2 Challenges of human occupancy counting with a single PIR sensor

Our aim is to obtain an accurate online estimate of the number of occupants in an office meeting using data from a single PIR sensor. The simplicity of the sensor will necessarily create some specific challenges that must be carefully considered when modeling the data. A PIR sensor outputs the change in temperature of a passing, heat-emitting object compared to the background temperature of the field of view, therefore we need to verify the sensitivity of our findings to the choice of a monitored room in which each experiment is performed. Since the human body is a heat-emitting 'object', occupants within the monitored environment can be easily blocked from the field of view of a single sensor by other occupants.

The sensor output staturates at a maximum value of 1.0 (see Section 6.2.1 for details), therefore there is a limited range of motion that we can actually differentiate with this type of sensor. For example, if two or more people are sufficiently active and close to the sensor to generate more than the maximum range of detectable motion, the sensor would be unable to detect the motion patterns of the rest of the occupants. That is, the occupants occlude each other not only by physically constraining the field of view of the sensor, but also by exceeding the maximum range of motion that the PIR can measure.

We notice that more occupants would on average generate increasing range of motion as long as we observe them for long periods of time. Therefore, a simplistic approach to estimate occupancy is to assume that occupancy count increases with the increase of motion. However, within short observation time windows (e.g. 30 seconds or less) it is likely that the temporally local behaviour of particular individuals will undermine this assumption. This is why we need to carefully handle such temporally local behaviours to extract properties of the global behaviour of interest, whether a participant is in the room or not.

6.1.3 Related work

Occupancy counting in an environment is a crucial task in human behaviour sensing and as such it has been widely studied. Yet, it is typically approached by employing either a occupancy count sensor that covers the entire area of interest, or keeping a tally of people entering and leaving at all entry and exit points. While the first approach is generally more accurate, the higher price and energy consumption of these systems makes them prohibitive for many real-world applications.

1. Person-counting sensors that cover the entire area of interest usually consist of high resolution video, stereo cameras and thermal imaging devices. A tracking algorithm is used to count human bodies from the image, for example by using supervised machine learning from dot-annotated images (Lempitsky & Zisserman, 2010), or using head-detection algorithms from stereo camera images (Van Oosterhout *et al.*, 2011). Chan & Vasconcelos (2012) used unsupervised machine learning to segment components of homogeneous motion before applying Bayesian regression, and this approach shows promising improvements for locating and counting people in crowded places from video data. Yang *et al.* (2003) proposed a real-time network that does not depend upon object tracking, which makes the scheme much less computationally prohibitive; the high cost of the data acquisition device still remains an issue though.

2. Considerable effort has been invested in trying to avoid the need for expensive devices. Most progress in that direction is obtained by systems that rely on counting at all entry and exit locations of a closed environment (Hashimoto *et al.*, 1997; Zappi *et al.*, 2007). For example Zappi *et al.* (2010); Yun & Lee (2014) placed three PIR sensors in a hallway to identify direction of movement and relative location of people passing. Agarwal *et al.* (2010) instead combined PIR sensors with reed switch door sensors for occupancy counting with the purpose of optimizing the energy consumption of an office building. Wahl *et al.* (2012) presented a similar approach, but using only PIR sensors at all entries and exits.

3. Alternative systems use multiple low-price sensors at different locations which are tied through a probabilistic model that combines information from the different outputs (Khan *et al.*, 2014). Dodier *et al.* (2006) used a probabilistic belief network to model occupancy based on data from multiple PIR sensors (4 PIR sensors per room) placed on the walls rather then entry/exit locations. This method assumes that the number of occupants is constant over time and that the system can be trained on typical behaviours common for the monitored room. The belief network is calibrated on historical data for the monitored rooms and does not adapt after the training stage, which makes the system highly dependent on the historical data and sensitive to non-observed behaviour. Lam *et al.* (2009) used HMMs to quantify occupancy count from extracted features of multiple types and locations of energy-efficient sensors. This approach shows average accuracy of 80% in open-plan buildings, where each occupancy estimate is based on a window of one minute sensor data. The accuracy is measured as the number of correctly estimated intervals of a minute divided by the total tested time(in terms of number of minute intervals).

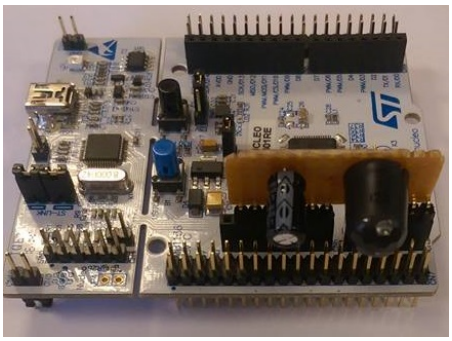
4. Assuming resources are unconstrained, perhaps the most accurate occupancy count can be obtained from systems that make use of both expensive sensors to monitor larger areas of a building, multiple motion based sensors monitoring each entrance and exit, historical data of building occupancy, CO_2 sensors and smoke detectors etc. (Meyn *et al.*, 2009; Erickson *et al.*, 2009). In the simplified case of single room monitoring, much research has been directed towards systems with diverse sets of sensors that are able to infer comprehensive human activity (Kientz *et al.*, 2008; Brooks, 1997), but the focus of such systems is behaviour modeling rather then occupancy counting, and these systems therefore have high complexity and cost.

In contrast, we suggest using a single PIR sensor and flexible probabilistic model to model simpler behaviour that are closely related to the number of attendees.

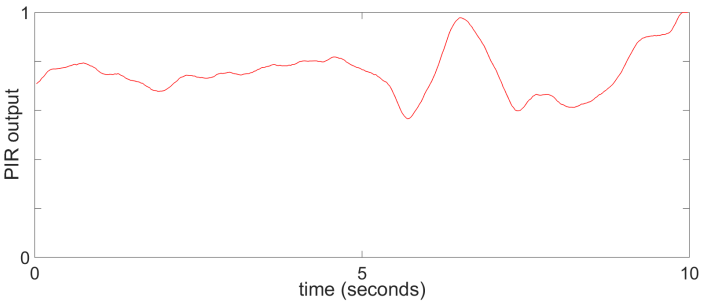
6.2 Experimental Setup

6.2.1 Collection devices

In this study we attached a single PIR motion sensor (Panasonic NaPiOn series AMN21111) on a printed circuit board (PCB) to an ARM mbed NUCLEO F401-RE microcontroller board, powered through a USB cable that connects it to a laptop. The PIR sensor is an analog output sensor as opposed to digital output ones in NaPiOn series. It is a standard type PIR, 14.5mm tall, lens surface area 9.5mm and 9.8mm mounting hole. It has 5m detection range, horizontal view of 82^o and top (vertical) view of 100^o and records approximately 30 single dimensional digital measurements per second (30Hz sampling rate). The PIR is connected to the 12-bit ADC embedded in ST Nucleo-F401-RE microcontroller as shown in Figure 6.1(a). We use the mbed compiler to read the analog values from the PIR. The mbed compiler uses a function to convert analog values



(a) Microcontroller board



(b) Raw sensor output

Figure 6.1: *Fig(a)*:Raw digital data recorded using the standard digital PIR sensor for 10 seconds. *Fig(b)*:Image of the data acquisition board consisting of a NUCLEO F401-RE mbed microcontroller board with a single PIR sensor on a PCB connected to the ADC port of the microcontroller through the Arduino connector.

to digital in a range from 0.0 to 1.0 where 0.0 represents 0 volt while 1.0 represents *voltage drain drain* (Vdd). Analog values between are represented by a floating-point number between 0.0 and 1.0. The board is placed in the middle of the room, adjacent to the wider wall in rectangular rooms, with the sensor facing the room interior. The height at which the sensor is positioned varies between 0.70m and 1.00m. The analog data generated by the sensor is sent to the ADC inside the microcontroller that converts analog data to digital, which is then transmitted to the laptop through the USB interface for further processing.

Figure 6.2: Example of a monitored room with no occupants inside. The board is placed in a typical position in the middle of the room at 1m height.

6.2.2 Data collection

The data acquisition board is deployed in 7 different conference rooms (see Figure 6.2) in an office building, where the rooms vary in dimensions, access to sunlight and maximum occupant capacity. Data has been collected from randomly chosen real meetings in the ARM Corporation headquarters in Cambridge, UK and there is variation in the number of individuals and the nature of each meeting. The monitored meetings involved white board sessions; seated formal meetings; slide presentations; shared conference calls etc. The board was carefully placed in the middle of the room, in order to maximize the PIR sensor coverage. Upon the start of each meeting sensor data was recorded where the first and the last five minutes of the recorded PIR sensor data are removed to account for the system installation and occupants to settling in. Note that the start and the end of a meeting can be automatically detected from the PIR output with great accuracy (see Figure 6.3).

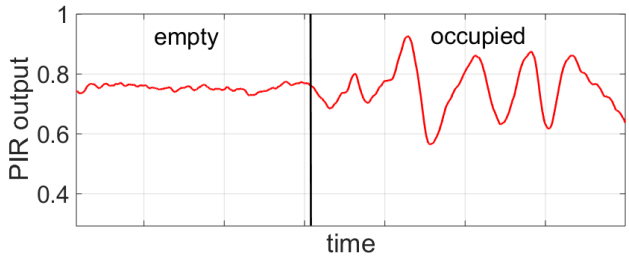
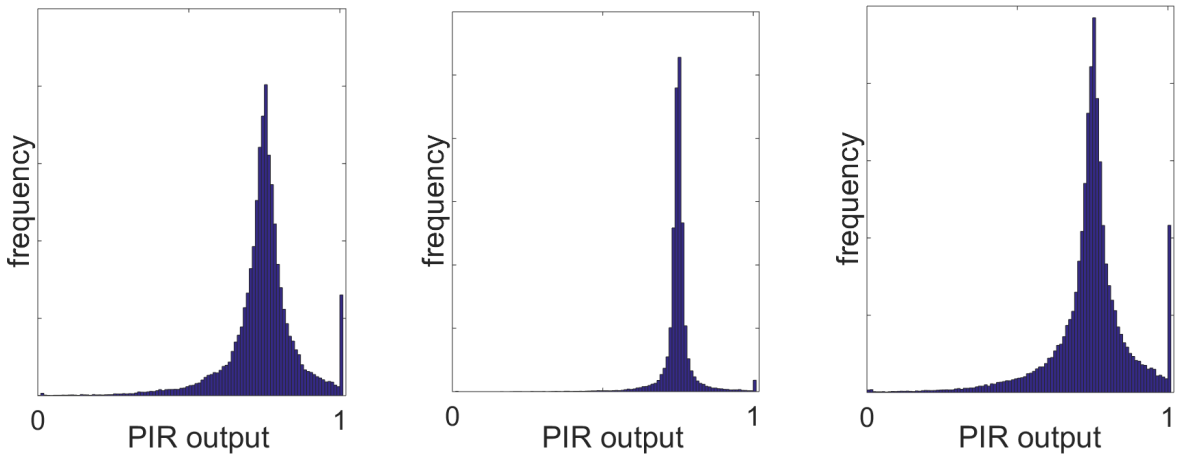


Figure 6.3: We demonstrate the obvious change in the PIR output as soon as a monitored room becomes occupied. The PIR output is for approximately 30 seconds and the black verical line sinifies the moment when individuals start entering.

6.2.3 Sensor data description

The analog output of the PIR sensor is converted (by the 12 bit ADC) to real numbers in the range of 0.0000 to 1.0000 with 4 decimal places. The temporal fluctuations in this signal reflect certain movements in the monitored environment (Figure 6.1(b)). When there is a lot of movement in the room, the PIR analog output reaches the maximum value, which, in turn, is converted to 1.0 by the ADC. The challenge we are addressing entails analyzing these fluctuations to infer the number of people occupying the monitored room. The PIR output for a typical 1 hour meeting comprises a set of approximately 120,000 floating-point numbers.

Figure 6.4 depicts the statistical distribution of the sensor data from different meetings, which ignores the time ordering of the data. The sharp peak in the distribution at the median value, combined with the heavy tails and the truncation at the maximum ADC output 1.0 suggest that for longer durations the PIR data is well described by a mixture of a truncated *Laplace* distribution[2] centered at the median value, and a Dirac delta distribution centered at 1.



(a) 9 people (b) 2 people (c) 5 people

Figure 6.4: Histogram of raw PIR data from three different meetings with varying number of occupants and approximately 1 hour duration.

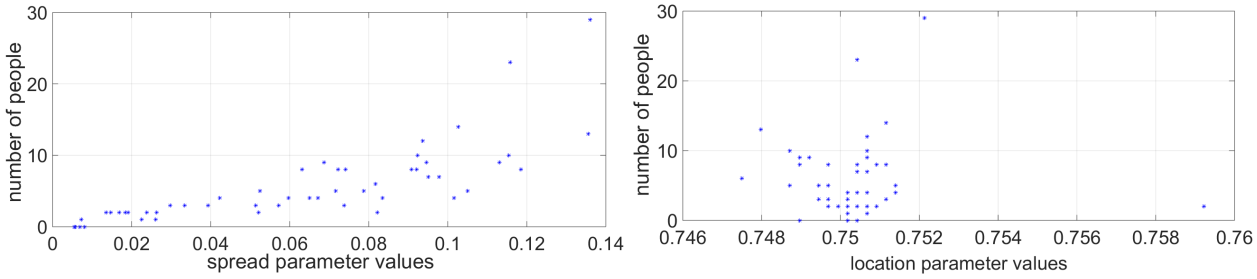
[2]The Laplace distribution governs the difference between two independent identically distributed exponential random variables. The probability density function of a Laplace random variable $X \sim \mathcal{L}(\mu, b)$ can be written as: $p(X = x \,|\, \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$ where μ is a location parameter estimated with the sample median and b is called diversity or spread parameter of the Laplace distribution.

6.3 Laplace modeling

We showed that the PIR output for long segments of different meetings is well described with a mixture of a truncated Laplace distribution and a Dirac delta distribution centered at 1.0. Ignoring the delta distribution, it is then reasonable to model the data from different meetings with different Laplace distributions, $\mathbf{x}_j \sim \mathcal{L}(\mu_j, b_j)$ where $\mathbf{x}_j = x_{j,1}, \ldots, x_{j,N_j}$ denotes the sensor data stream of PIR measurements from meeting j collated into a single vector, and (μ_j, b_j) are respectively the location and spread parameter of the Laplace. We estimate $\mu_1, \ldots, \mu_J$ and $b_1, \ldots, b_J$ using:

$$\mu_j = \text{median}\left(x_{j,1}, \ldots, x_{j,N_j}\right), \; b_j = \frac{1}{N_j} \sum_{i=1}^{N_j} |x_{j,i} - \mu_j| \tag{6.1}$$

for $j = 1, \ldots, J$, where J denotes the number of training meetings (in this study $J = 53$) and N_j denotes the number of PIR output points for meeting j. In Figure 6.5 we plot each μ_j and b_j against the number of people that have been present at meeting j. While the location parameters do not vary substantially across meetings, we observe that meetings with higher occupancy are indeed more likely to have larger spread (larger values of b), as expected. In addition, we notice that the relationship between the count and the spread parameter changes quite substantially for meetings with more than about 8 occupants. Examination of the monitored rooms shows that assuming normal seating patterns 8 occupants are the most that can fit within the field of view of the standard PIR type sensor without occupants occluding each other. We believe this is a limitation of the monitored environment and the position of the PIR sensor, rather than our proposed counting algorithm.



(a) Laplace spread

(b) Laplace location

Figure 6.5: Number of occupants for different meetings plotted against the Laplace parameters.

6.3.1 Regression component

The small amount of regression data coupled with the single predictor variables make *generalized linear models*[3] (GLMs) an appropriate parsimonious choice for modeling the dependence between the Laplace spread parameter and the occupancy count. We will treat low occupancy meetings with fewer than 8 occupants separately from the ones with 8 or more occupants, where for most practical purposes we need an unsupervised way of switching between those two regressions. The easier, but less accurate approach would be stratify at a hard value of $b \sim 0.09$ (which is a specific value estimated from the training data). Alternatively, we can

[3] A generalized linear model is a simple regression model which assumes that we can predict the expected value of some outcome variable$Y$ (which has some exponential family distribution) in terms of some predictor variables X using: $\mathbb{E}[Y] = g^{-1}(X\beta)$ where β are some unknown parameters of the model and $g(\cdot)$ is called a link function, which is usually determined based on the distribution of Y. The GLMs are largely used in practice because they are simple to use, fast to train, have relatively few parameters and provide intuitive inside about the relationship between the predictor variables and the outcome.

use a probabilistic switching mechanism once the estimated b enters an interval of uncertainty between the different models, or sacrifice some of the accuracy and use a single regression model.

For the data from both more and less occupied meetings multiple types of GLM regression were compared in terms of *mean absolute error* [4] (MAE); the best fit for meetings with up to 7 occupants is obtained with a linear model with Gaussian outputs; for the second strata of high occupancy meetings a log-linear model with Poisson outputs provides the best fit scoring the lowest MAE of the tried regression models(as for high occupancy meetings the number of occupants increases exponentially with the spread parameter).

The MAE for the low occupancy strata (less than 8 individuals) is less than 1. This suggests that with Laplace parameters estimated from the PIR data from an observed meeting, we can identify the number of occupants to within ± 1 individual on average. For the high occupancy strata the count prediction accuracy is reduced, but some relationship can be captured with MAE of the log-linear model less than ± 1.25 individuals.

While there exist much more complex regression models which could be used, for example support vector regression (Smola & Vapnik, 1997), kernel regression (Fan, 1992), Gaussian process regression (Williams, 1998) or regression based on convolutional neural networks (Bishop, 1995), they require substantial amounts of memory, computational power and training data. In addition, such methods trade interpretability of the classifier for empirical accuracy. More specifically, they often have large numbers of parameters and it is extremely difficult to predict from an analysis of the trained model what the effect on the occupancy count prediction will be when varying any one of these parameters. For example, support vector regression requires that all support vectors are held in memory, and requires *quadratic programming*[5] to train the regression model (Smola & Vapnik, 1997). Similarly, while *convolutional neural networks* have been used to solve difficult regression problems to high prediction accuracy (Bitvai & Cohn, 2015; Kang *et al.*, 2014; Pathak *et al.*, 2015), these require vast amounts of training data and computational power which makes them generally out of reach of low power embedded microcontroller systems.

By comparison, parameter training in GLMs leads to convex optimization problems which can be optimally solved using simple gradient descent algorithms.

6.3.2 Time window duration

The Laplace spread parameters for each meeting were estimated from all of the PIR data for that meeting, most often approximately an hour. Therefore, to make a prediction for the occupancy count, we have to wait the whole duration of the meeting. To be practical, the system needs to be able to work for much shorter time windows.

We next investigate this by fitting a Laplace distribution to shorter time segments of the raw sensor data. Instead of estimating parameters from the data for the whole meeting, we estimate the same Laplace parameters for every 2 minutes time windows, that is we partition each meeting in multiple smaller, non-overlapping time windows. The problem we will face is that shorter time segments of PIR data are more conflated with short-term individual behaviour which is not representative of the current number of occupants. Figure 6.6 shows estimates of the spread parameters evaluated every consecutive 2 minutes of a meeting with 9 occupants present for the entire duration.

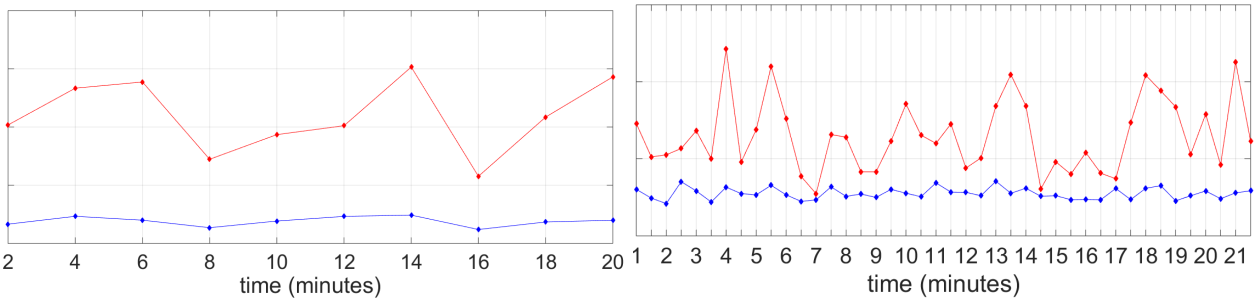
Ideally, the Laplace parameters would be almost constant across all time windows, indicating that data recorded from the same meeting is summarized with the same parameter values. The varying spread of the raw data from the same meeting is explained by the varying movements of the occupants in that duration.

[4] MAE is the average difference between the estimated and the true value of an outcome variable.

[5] Quadratic programming is the mathematical optimization problem of optimizing a quadratic function of several variables subject to linear constraints on these variables. It has been proven that in general this optimization problem is NP-hard.

This variation will be due to temporally local and/or individual behaviours which depend upon the precise nature of the meeting and the motions of the occupants during the window, the effect of which diminishes over longer time windows.

We address this problem of the Laplace parameters varying during the meeting by clustering the training data into groups of similar motion patterns and then matching the motion structure discovered onto patterns of human behaviour we expect to observe. To have a sufficiently flexible grouping of behaviour, and to allow the number of observed behaviours to grow as more data becomes available, we model these groupings using the iHMM (see Chapter 5). In this way, instead of using all the PIR data we focus the analysis only on the clusters that are most universally likely to describe the occupancy count. This approach substantially reduces the variation in spread parameters over the duration of the meeting, for shorter time windows (Figure 6.6).



(a) 2 minutes windows

(b) 30 seconds windows

Figure 6.6: Laplace spread parameters for different time windows of a meeting with 9 occupants. The red line shows the spread parameter estimated from all the raw PIR sensor data, whereas the blue line shows the spread parameter estimated from only points in the selected small motion behaviour.

6.4 Extracting behaviour from PIR data

In Figure 6.7 we show the clustering produced with iHMM of 15 seconds PIR signal, where different colors denote different clusters. The iHMM has been trained on all of the training data which exceeds 53 hours rather than just on the 15 seconds that are displayed.

Lets assume T denotes the number of PIR recordings from all meetings, using the notation from Section 6.3 this implies $T = \sum_{j=1}^{J} N_j$. Then we concatenate the series of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_J$ into univariate time series $x_1, \ldots, x_T \equiv x_{1,1}, \ldots, x_{1,N_1}, \ldots, x_{J,N_J}$. Every observed PIR output x_t with $t \in \{1, \ldots, T\}$ is associated with a hidden variable z_t indicating the state of that observation and every represented state $k \in \{1, \ldots, K\}$ is modeled with a Laplace distribution $\mathcal{L}(\mu, b_k)$ with fixed location parameter μ and cluster specific spread b_k ($x_1, \ldots x_T$ is modeled with HDP-HMM with Laplace distributed components). Conveniently, the Laplace distribution with fixed location has a conjugate prior. Further we observed similar location parameter values across different meetings supporting our assumption of fixed μ. The conjugate choice for prior over the spread parameters b_k is the inverse-gamma distribution, $b_k \sim \text{InvGamma}(\nu_0, \chi_0)$[6].

By fitting an iHMM to the raw PIR data, we aim to group together segments of the time series that are similar. In this way, observations that are coupled into the same state are more likely to describe the same physical pattern of movement. Note that typical human behaviours (e.g. walking, sitting down, standing up) are complex and so are composed of many different types of motion. Without making restrictive assumptions

[6]The inverse-gamma distribution is the distribution of the reciprocal of a gamma distributed variable. Inverse-gamma random variable $X \sim \text{InvGamma}(\nu, \chi)$ has a probability density function: $p(X = x\,|\nu, \chi) = \frac{\chi}{\Gamma(\nu)}^{\nu} x^{-\nu-1} \exp\left(-\frac{\chi}{x}\right)$.

about the movement described by the recorded PIR signal, we are more likely to group together similar types of motion rather than composite human behaviours. At the same time obtaining the structure of the observed motion patterns is key to understanding how the different human behaviours are formed and in what way those behaviours differ based on the sequence of movements that form them. For our problem of occupancy counting, mapping sequences of motion patterns into composite behaviours is not the focus of the problem. However, HMMs have already proven useful for the more difficult problem of tracking the activity of monitored individuals in different problem domains (Mannini & Sabatini, 2012; Toreyin *et al.*, 2008).

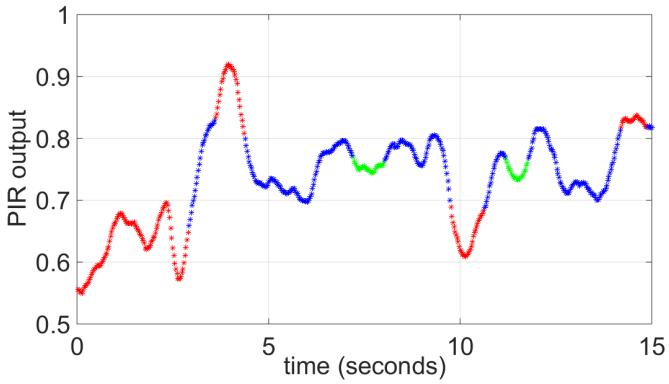


Figure 6.7: Clustering output of the iHMM applied to 15 seconds PIR data. Different colors denote different clusters (states); the number of the clusters has not be specified a priori, but learned from the data.

We describe the collection of motion patterns that describe small movements as 'small motion behaviour'. In order to separate parts of the PIR output describing this small motion behaviour, to filter out larger movements, we examine the PIR recordings from an empty room. More precisely, we examine which states occur for empty rooms once the iHMM is fitted to whole of the data. Then we identify segments of the PIR signal, from occupied rooms, that are grouped together under clusters found in non-occupied rooms. The iHMM groups together motions that are temporally similar, so larger movements would be coupled in separate groups and we can easily filter them out. Large temporally local fluctuations in the PIR output reflect some temporally local human behaviour and will bias the occupancy count estimate (unless more information is available about the nature of these behaviours). In Figure 6.7 the blue and green states are motions describing the 'small motion behaviour' and the data belonging to the red cluster is filtered out. Note that red state groups both data with bigger and smaller PIR output, as points are grouped with respect to their common spread and time dynamics rather than absolute value. By focusing only on specific states, we are comparing Laplace parameters estimated from comparable (similar) sequences of PIR data which will make our estimates of those parameters less variable and more robust to reducing the occupancy count estimate time window (c.f. Figure 6.6). In the collected training data approximately 70% of all of the training data groups into small motion behaviour and the remaining 30% of the PIR output is filtered out.

Filtering out undesired motion behaviours and reducing the observation time windows will also help us exploit a more accurate switching mechanism between the two regression models for more and less occupied environment when needed. One efficient way we suggest for that switching would be to specify an interval of uncertainty for b; estimated b falling inside of that interval would imply uncertainty about which regression model to use. Whenever values for b estimated from the latest PIR output are from the uncertainty region, we do not choose a regression model but we proceed by estimating another b from the next window. We repeat that step if needed and based the first value of estimated b and also obtained sequence of values b so far, we choose with higher certainty the appropriate regression model to output the occupancy count. Using

this mechanism can cause certain delays in the estimation output, but delays rarely exceed 2, 3 times of the aimed time window duration (for example delay of 1 minute and 30 seconds instead of the assumed 30 seconds windows for estimation).

6.5 System overview

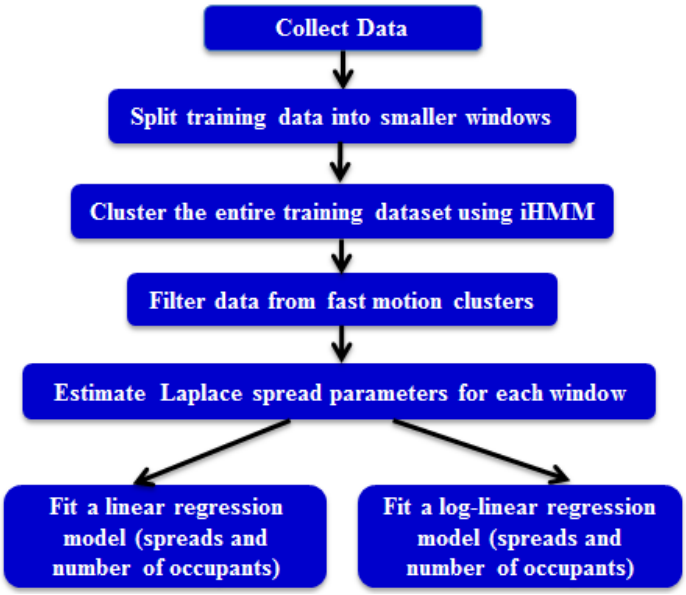


Figure 6.8: Architecture of a novel occupancy estimation system.

In this section we describe the key stages in our proposed solution to occupancy estimation (Figure 6.8). The data acquisition process and the statistical nature of the recorded PIR output is discussed in detail in the experimental setup. We split the training data into different time windows of PIR output to examine the duration of signal sufficient to accurately estimate number of monitored people. Once data has been partitioned to smaller time windows, the training data is coupled using iHMM in order to extract physical behaviour of interest from the raw PIR signal. The behaviour that biases the occupancy estimation is filtered and we model the remaining data using Laplace distribution. The estimated Laplace parameters describe well how populated a meeting has been and can be efficiently used in a regression model. Different regression models are used for more and less occupied meetings to maximize estimation accuracy.

6.6 System evaluation

In this "in-the-wild" study we recorded PIR sensor data from 53 real-life meetings, 37 of those had up to 7 participants, the remaining 16 had more than 7, and the 2 most occupied meetings had 23 and 29 occupants, all with different meeting durations. The two most occupied meetings are excluded from the analysis, because the data recorded from such over-populated meeting rooms (considering the size of the conference rooms) is not meaningful. Indeed, the maximum seating space in the biggest of the monitored rooms is 14 people and typically exceeding this capacity leads to severely limiting the field of view of the PIR sensor which causes severe sensor occlusion and irretrievably biased sensor output.

We recorded simply the PIR sensor output and the true number of occupants for each meeting, so the

Table 6.1: Percentage of time windows across all meetings (with less than 8 people) where the predicted number of occupants is within ±1 of the true number of occupants. In the square brackets is the percentage of time windows where the predicted number of occupants is within ±2 of the true number of occupants.

	Raw data	Small motion behaviour
30 seconds	63% [93%]	80% [96%]
1 minute	80% [93%]	83% [97%]
2 minutes	82% [96%]	85% [99%]
20 minutes	89% [97%]	92% [97%]

study has been highly non-invasive. The data from each meeting is split in small observation windows in order to track how the accuracy of the occupancy count system changes with the count estimation time window. Note that if we wish to receive an estimate of the current room occupancy every 30 seconds, naturally the accuracy of that estimate would be lower than an estimate obtained every 2 minutes or every 20 minutes. For numerous applications an occupancy count estimation updated only every 20 minutes would not be of great value so there is an inherent trade-off between accuracy and count estimate time window. We investigated time windows of 20 minutes, 2 minutes, 1 minute and 30 seconds. Additional investigation showed that processing windows longer than 20 minutes does not appear to provide a substantial increase in occupancy count estimation. We still treat low and high occupancy count meetings differently in the analysis due to the different statistical nature of the data in these different occupancy strata.

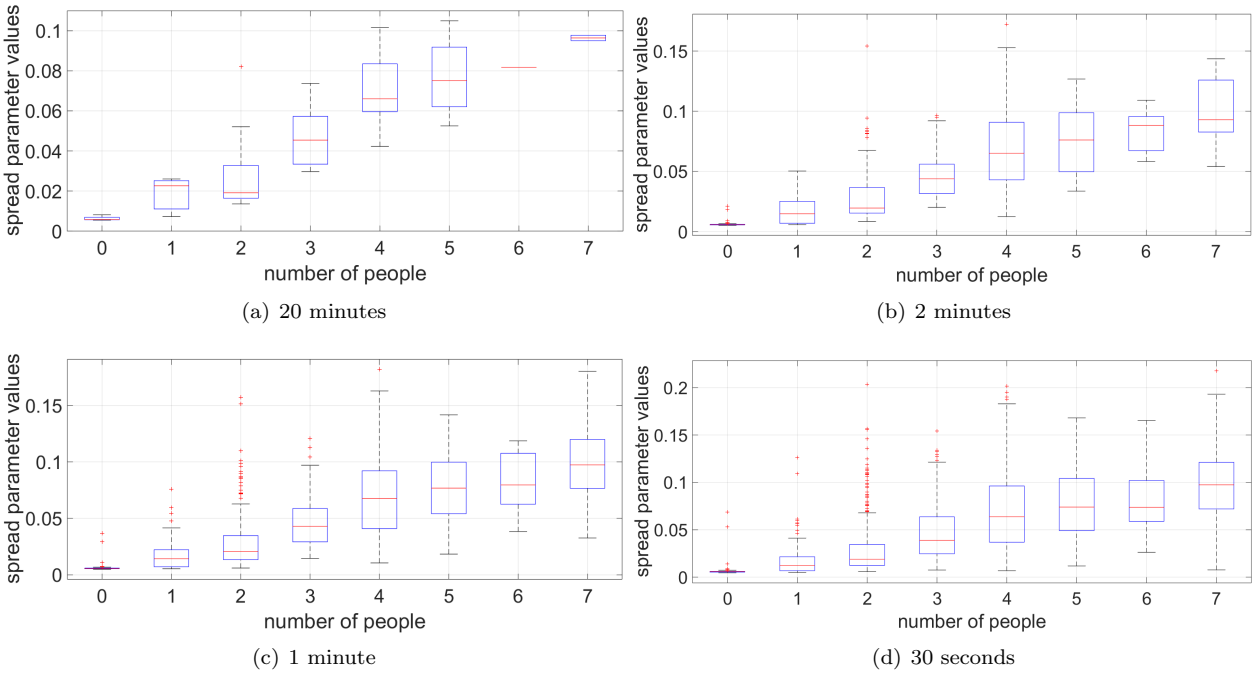
6.6.1 Fewer than 8 occupants

In the case of a small numbers of occupants, a linear Gaussian regression model performed best in terms of MAE and is used to predict the human occupancy count from the spread parameter alone (Figure 6.9). For shorter estimation time windows, the relationship between occupancy count and PIR data spread becomes unclear, the effect of which is clear from the numerical prediction accuracy estimates for predictions within ±1 and ±2 (Table 6.1) of the true occupancy count.

After the raw data is segmented with the Laplace iHMM and the data in the states describing large movements is filtered out, we estimate spread parameters only from the remaining data representing small motion behaviour. Following the same recipe, Gaussian linear regression is used to predict occupancy count from the "stabilized" Laplace spreads for different count estimation time windows (Figure 6.10 and second column of Table 6.1). The resulting overall increase in prediction accuracy confirms the positive effect of iHMM behaviour segmentation.

Table 6.2: Percentage of time windows across all meetings (with at least 8 people) where the predicted number of occupants is within ±1 of the true number of occupants. In the square brackets is the percentage of time windows where the predicted number of occupants is within ±2 of the true number of occupants.

	Raw data	Small motion behaviour
5 minutes	68% [79%]	59% [86%]
20 minutes	79% [84%]	71% [84%]



(a) 20 minutes

(b) 2 minutes

(c) 1 minute

(d) 30 seconds

Figure 6.9: Box plots of Laplace spread parameters estimated from raw PIR data for different meetings with up to 7 occupants, over different estimation time window durations. Top and bottom edges of each blue box are 25th and 75th percentiles respectively, the middle red line is the median, red pluses denote outliers.

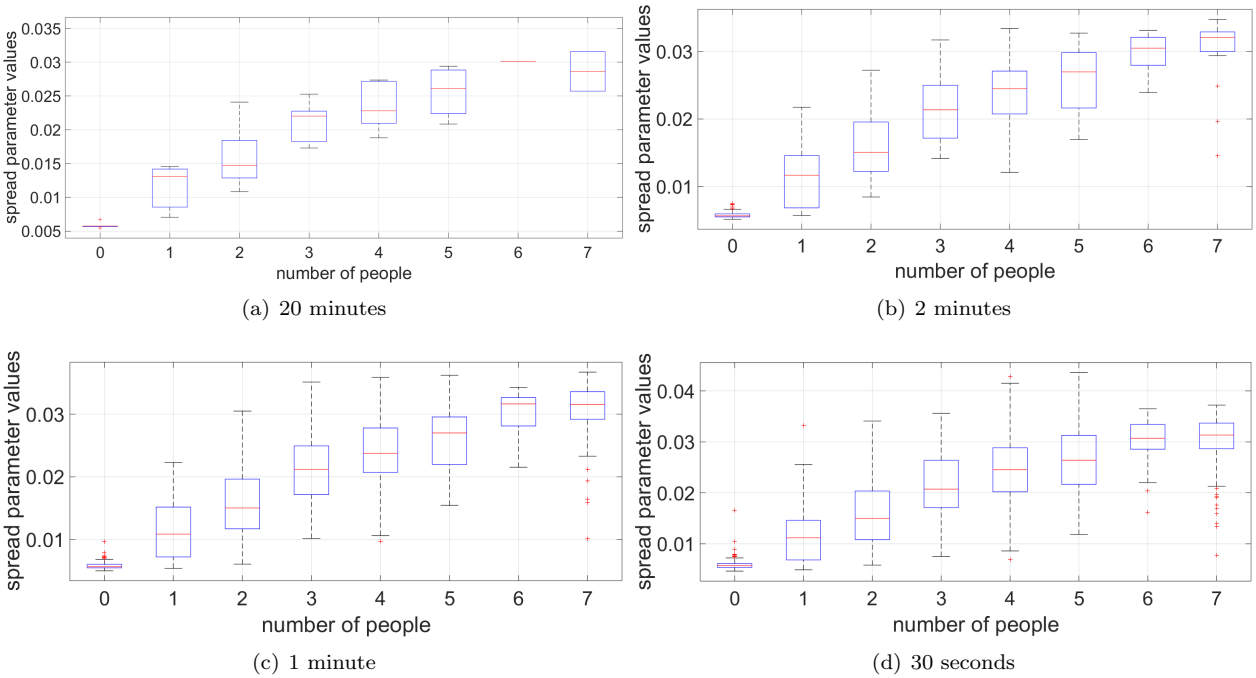
6.6.2 At least 8 occupants

For larger numbers of occupants which occlude each other, a Poisson log-linear regression model is found to provide the most accurate predictions (Figure 6.11). The predictive power of the Laplace parameters reduces significantly in this high occupancy strata due to the reasons discussed above and the error of this approach for windows smaller than 5 minutes is substantial. In addition, the benefits of behaviour extraction stage are diminishing and regression on both Laplace parameters evaluated for both the raw data and "stabilized" Laplace parameters performs almost equally in terms of MAE. The estimation accuracy within ± 1 person and within ± 2 can be found in Table 6.2.

6.7 Computational efficiency

6.7.1 Choice of inference algorithm

Practical applications of sophisticated Bayesian probabilistic models have been few largely due to the complex and computationally demanding inference algorithms involved for learning the parameters of such models. As a BNP probabilistic model for time series data, the iHMM is no exception and careful consideration is needed to choose fitting procedures which are tractable for implementation in low-power embedded microcontroller hardware. The system should be able to execute in a resource constrained embedded environment after deployment where a microprocessor would take the data from the PIR sensor and run segmentation algorithm on board. In this study we compare several different iHMM inference algorithms: the beam sampler (Section 5.3.1), direct assignments Gibbs sampler 5.3.1 and iterative maximum a posteriori (MAP) inference Section 5.4.2 and we display their performance in Table 6.3. Note that we are less interested in the quality of fit of



(a) 20 minutes

(b) 2 minutes

(c) 1 minute

(d) 30 seconds

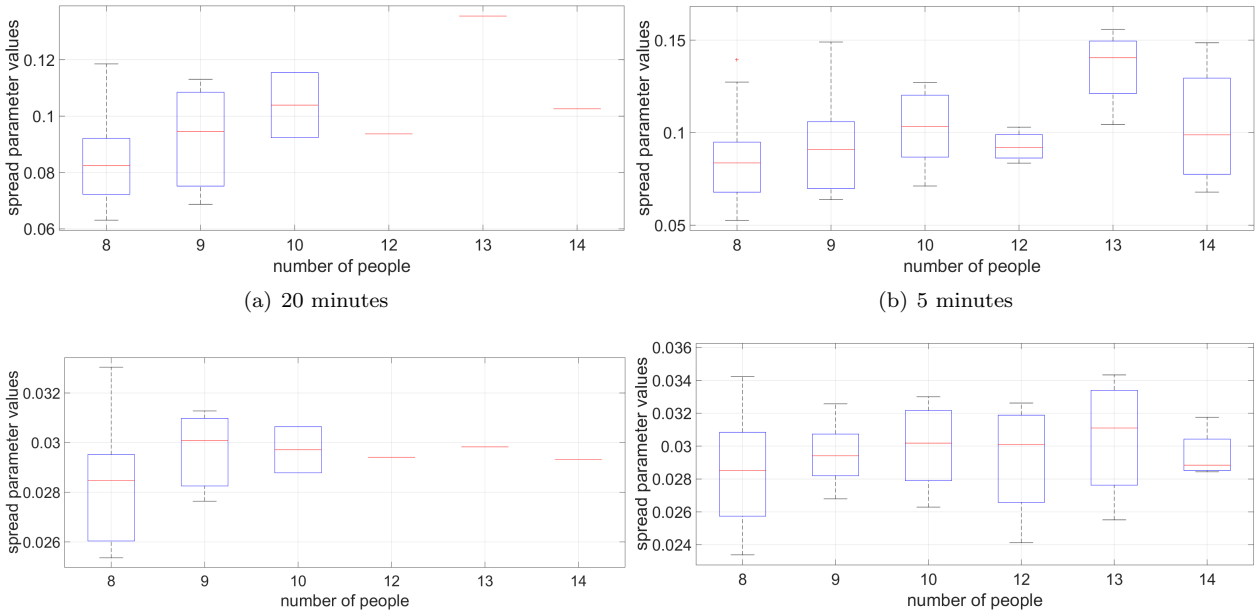
Figure 6.10: Box plots of the "stabilized" spread parameters estimated only from the saddle behaviour clusters across all meetings with at most 7 occupants.

Table 6.3: Mean absolute error (MAE, interquartile range in brackets) as a measure of occupancy count prediction accuracy using "stabilized" Laplace parameters from PIR data only for small motion behaviour clusters. Each column corresponds to iHMM clustering performed using a different inference algorithm. The last row shows speed comparison in terms of iterations to convergence.

	Beam sampler	Gibbs sampler	Iterative MAP
30 sec.	0.95(0.7)	0.98(0.8)	0.99(0.8)
1 min.	0.87(0.7)	0.89(0.8)	0.91(0.7)
2 min.	0.79(0.6)	0.81(0.7)	0.84(0.7)
20 min.	0.64(0.6)	0.72(0.6)	0.70(0.7)
Iterations	125	100	6

the iHMM to the raw PIR sensor data than the prediction error of the regression component of the system using estimates of the "stabilized" Laplace parameters obtained using that iHMM, where the parameters have been estimated using different iHMM inference algorithms. This is because ultimately we care about accurate human occupancy counting rather than learning the iHMM per se. We report the iterations that each inference algorithm required to convergence where an iteration consist of a full sweep through the training data and the model parameters. Computational cost of a single iteration across algorithms is not equivalent, but for the chosen application is comparable.

Theoretically, both beam and Gibbs sampler inference algorithms are guaranteed to converge on the optimal iHMM fit eventually. However, the stochastic nature of both samplers makes them highly computationally demanding and they can easily take two orders of magnitude more iterations to converge than iterative MAP. At convergence both stochastic algorithms will generally outperform iterative MAP in terms of iHMM parameter estimate accuracy, but we observe that the improvement due to better iHMM parameter estimates does not translate into sufficiently improved occupancy count to justify such large increase



(a) 20 minutes (b) 5 minutes

(c) 20 minutes (d) 5 minutes

Figure 6.11: Box plots of the spread parameters over all meetings with number of occupants varying between 8 and 14.

in computational effort over iterative MAP. Indeed, iterative MAP is simple enough that it can be used where computational resources are at a premium, as would be the case for our experimental setup using a microcontroller board.

In practice, it is most meaningful to train the system before deployment on powerful hardware using the most accurate approach (e.g. the beam sampler) and then proceed by saving some trained representation of the iHMM and using iterative MAP method for newly arriving windows of PIR data after deployment.

6.7.2 Resource evaluation

After testing the empirical accuracy of the proposed system and evaluating performance of different iHMM inference methods for the segmentation stage, we proceed with a feasibility study for potential deployment of the proposed system. We first train an iHMM on PIR data from 50 collected meetings using beam sampling. The model parameters consisting of the transition matrix π, component mixing weights β and the Laplace spreads b are stored in a device's memory (those take floating point values and for 50 meetings take 11.2 KB of memory). The system is deployed and a modified MAP-iHMM or equivalently dynamic MAP-iHMM are used *online* to segment incoming streams of PIR output (storing only 30 seconds of data at a time). After each 30 second window, the model parameters are updated accordingly, to incorporate information about the latest data and prediction of the level of occupancy is made based on some pre-trained regression models.

The feasibility performance of the system after deployment (with iterative MAP method on board) is tested on two different microcontroller units (MCUs): Nucleo F070-RB and Nucleo F401-RE. They are equipped respectively with ARM Cortex-M0 and ARM Cortex-M4F CPUs (full specification of the boards in Table 6.4).

Table 6.4: Specification of the MCUs. Memory is measured in terms of SRAM size with the flash size is in the brackets.

Board name	CPU	Clock (MHz)	Memory (KB)	Floating point unit (FPU)	Peripherals
Nucleo F070-RB	Cortex-M0	32	16 [128]	No	ADC, DMA, TIM3
Nucleo F401-RE	Cortex-M4F	84	96 [512]	Yes	ADC1, DMA2, TIM2

Table 6.5: Computation time and SRAM memory footprint of the system ran on difference MCUs

Board name	Execution time (seconds)	Memory Requirement (KB)
Nucleo F070-RB	95	11.9
Nucleo F401-RE	1.15	11.9

Memory usage and real time performance To evaluate computational demand, we use execution time as a metric, and test how long each MCU takes to calculate the estimated occupancy count, after collecting 30 seconds of data. The significantly slower Nucleo-F070RB takes approximately 95 seconds to execute, which would not fulfill the requirement for the system to be ready in 30 seconds to receive the new coming stream of PIR output. The execution time for the higher performance Nucleo F401-RE is reduced to 1.15 seconds due to its faster clock speed and its single-precision FPU which allows to quickly perform floating-point operations. If we recompile with the floating-point instructions being emulated, the performance of the Nucleo F401-RE is still acceptable with 9.55 seconds execution time. The memory footprint of the iterative MAP running on the 30 seconds stream of data is 0.7KB (for both MCUs) and we summarize the results in Table 6.5.

Power consumption and battery lifetime We estimate the power consumption and battery lifetime of the MCUs with attached PIR sensor assuming it continiously estimates occupancy using iterative MAP: the average power consumption of Nucleo F070-RB is 8 mW and the average power consumption of Nucleo F401-RE is 18.7 mW. Consider now that boards are deployed in a building to run indefinitely, supported through an external battery connected to the MCUs. We test two type of LiPo(lithium-ion polymer) batteries – one with large capacity of 2200mAh and one with smaller capacity of 120mAh, both operating at 7.4V. The results are summarized in Table 6.6. The slower Nucleo F070-RB can function 5 days with 120mAh battery and 86 days with 2200mAh one, without replacement or charging. The faster board Nucleo F401-RE on the other hand can function 2 days with 120mAh battery and 36 days with 2200mAh one, which can be considered quite a feasible performance. All of those figures are obtained assuming we update our estimate of the occupancy every 30 seconds. If that estimate is updated less often or segmentation is not performed on every window of PIR data, this would additionally increase the life time of both of the batteries.

Table 6.6: Power consumption and battery lifetime

Board name	Average Power Consumption (mW)	Battery Lifetime of 120mAh LiPo (Days)	Battery Lifetime of 2200mAh LiPo (Days)
Nucleo F070-RB	8	5	86
Nucleo F401-RE	18.7	2	36

6.8 Future work

Type of the PIR sensor In addition to the standard type PIR sensor, we also used a slight motion type[7] PIR (Panasonic NaPiOn AMN 22112 series) in the same experiments in order to validate the developed models for different PIR. The standard sensor seems to be more promising then the slight motion detector mostly due to the larger field of view. The slight motion sensor does not cover all of the monitored room with only 2m range and occupants seated in particular areas of the monitored room cannot be seen by the sensor. The accuracy of the occupancy count system would benefit from exploring additional types of PIR sensors with more sensitivity, range and wider field of view. Further, installing a second PIR sensor on the opposite side of the room and analyzing the output of the two jointly may help to address both the problem of mutual occlusion of the sensor by the occupants, and the problem of limited field of view.

Position of the sensor The data acquisition board was placed on a table positioned approximately in the middle of the room next to the wall. The table was part of the chosen office room furniture and as a result its height varied slightly in the different meeting rooms. The results did not seem to be influenced by the exact height of the table, but placing the sensor on one particular side of the room led to occupants occluding each other during more populated meetings (typically with 8 or more people). This problem can be easily addressed by testing different positions of the PIR sensor; a promising start would be the ceiling of the room. This would make the installation of the system more challenging, but it is likely to lead to a consistent improvement of accuracy due to the clear unobstructed view of all the occupants that the sensor will be afforded in this physical configuration. In addition, PIR sensors installed on the ceiling are likely to increase the maximum distinguishable occupancy count. The accuracy of the system and its invariance to sensor location and position can additionally be improved with more training data accounting for different physical configuration scenarios.

Behaviour modeling In the current implementation, the key assumption made is that small movement patterns will describe better the number of people in a room, as they are less intentional and are independent of the nature of the meetings. We expect this assumption to hold in most human counting scenarios in office environment and no additional information about the nature of the meetings has been incorporated. In effect, we have sacrificed some of the predictive accuracy to obtain a generally applicable human occupancy counting system. If we incorporate additional information about the behaviour of the monitored occupants and the nature of the meeting, this can potentially to improve our prediction. For example, if we assume that we are monitoring conference meetings with duration of 1 hour starting on the hour, we would immediately know that major occupancy changes occur only once every hour and we can use the whole hour PIR data to improve our predictions. Another example would be if we assume occupants enter the monitored environment through a single gate: in that case we would use the iHMM to segment the event of entry/exit of an occupant and later try to classify the difference in patterns between entry and exit of an individual.

Limitations The low dimensionality of the PIR sensor makes the system sensitive to occupants occluding each other from the view of the sensor. Even with optimized positioning, there is a fairly limited number of occupants that we can expect to monitor with single PIR. To monitor larger spaces, we would need to place multiple PIR sensors at the different parts of the room, so that all of the area is inside of the field of view of at least one PIR sensor. With few simple updates, the suggested method can be used to process data

[7] The slight motion detection type of PIR sensor is lined-up with special detection lenses for slight motion or narrow spot detection.

from multiple PIR sensors. The single PIR measurements would be replaced with multivariate ones having measurements from different sensors in each dimension. We would also add more predictors in the regression stage to incorporate the information gained from the different PIR sensors. If we want to deploy the system in hallways, cafeterias or other office facilities, substantial additional training and calibration would be needed. The behaviour extraction stage simply groups together similar motion behaviours, so we believe it is highly adaptable to different scenarios. However the assumption of which behaviours are most correlated to the occupancy count can change with the nature of the monitored activity and should be carefully re-considered for follow-up applications.

6.9 Discussion

The purpose of this study was to demonstrate the potential of using a single passive infrared (PIR) sensor for more complex tasks than motion detection. We demonstrate how such a simple sensor combined with "intelligent" machine learning models exploiting our novel iterative MAP inference algorithm, can be utilized to solve the more complex problem of counting occupants in a room. While the accuracy of the proposed system does not yet reach the current state of the art obtainable with stereo cameras and computationally demanding image processing algorithms (or multi-sensor devices), our approach shows the ability to count the number of room occupants to within ± 1 individual while substantially reducing the hardware costs, computational power and the need for specialist installation. Applications where accuracy is not critical, for instance, optimizing energy usage in buildings, can benefit from this cost-effective and easy to deploy approach. To our knowledge, the system discussed in this report is the first attempt at designing a human occupancy counting system using a single, low-cost PIR sensor.

Chapter 7

Conclusion

We conclude this thesis with a summary of the principal contributions presented in earlier chapters and a discussion on some open research questions which can be explored in future. Throughout the course of the thesis we have proposed and studied simple deterministic methods for inference in both parametric and nonparametric latent variable models. We demonstrated how with simple extensions to popular segmentation and clustering techniques we can overcome most of their drawbacks while still keeping simple and scalable algorithms. Unlike most of the existing methods for deterministic inference in latent variable models, the proposed iterative MAP methods protect most of the key features of the probabilistic models, enable out-of-sample predictions and allow for model selection.

Standard MCMC sampling methods can still outperform the alternative deterministic iterative MAP methods, but we demonstrate multiple applications where the clustering performance of the two is comparable and iterative MAP reaches a solution orders of magnitude faster. This makes iterative MAP techniques extremely practical and they can be used to significantly extend the range of applications of typically 'slow-to-fit' BNP models. Naturally, the computational speed up of the proposed techniques also leads to some limitations, mainly that iterative MAP converges to a local solution of the assumed model and so it does not infer its complete posterior distribution. The asymptotic guarantees of MCMC sampling methods are conditioned on unlimited computational resources and running time therefore not always useful in practice, however a potential practical drawback of the approximate iterative MAP methods can be poor performance whenever data consists of highly overlapping clusters or states. In such cases clustering might be inappropriate and more computationally demanding density estimation might be needed.

7.1 Summary

Vast numbers of practical pattern recognition problems can be completely or at least partially solved using clustering and the most widely used clustering algorithm in practice remain simple techniques such as the K-means algorithm. This has motivated us study techniques which are nearly as simple as K-means, but can overcome most of its drawbacks.

In Chapter 3 we have studied typical clustering problems which are often approached with K-means and propose simple alternatives: we re-derived a new version of K-means (Section 3.4) that handles better poor initialization and also converges a lot faster to a good local solution; we also showed how using a new K-means with reinforcement (Section 3.2.2) we can produce clustering that accounts for the density of the data clusters; we discuss the more rigorous MAP-GMM (Sections 2.3.4 and 3.4) which unlike 'K-means like'

methods enables dealing with non-spherical data and enables standard model selection and out-of-sample prediction.

In order to deal with clustering problems in which the number of clusters K is unknown or changing, we proposed novel MAP-DPMM algorithm which is derived by applying iterative MAP inference to DPMMs. We demonstrated that in terms of clustering performance the proposed MAP-DPMM (Section 3.8.2) and collapsed MAP-DPMM (Section 3.8.1) can often outperform more computationally demanding inference methods for DPMMs like Gibbs sampling and VB inference. Furthermore, MAP-DPMM was used as a building block for more complex probability models such as the linear random mixed effects model (Section 3.10.2).

In Chapter 4 we discussed the benefits of using HDP for clustering problems with multiple variables some of which are categorical. Efficient iterative MAP scheme was proposed for inference in HDP mixtures (Section 4.4.2) and we demonstrated its superiority compared to the SVA alternative, the HDP-means algorithm. Chapter 4 also served as an introduction to the more sophisticated models needed in Chapter 5 to construct BNP models for analysis of sequential data.

In Section 5.4.2 we applied iterative MAP inference to various constrictions of the iHMM to obtain the novel MAP-iHMM and dynamic MAP-iHMM algorithms. Both algorithms converge orders of magnitude faster than MCMC methods and overcome the common problem with iHMM of overestimating the number of states. We also showed that MAP-iHMM can be used as an efficient tool for state recovery of DNA copy numbers in genomic hybridization and as a pre-processing tool for automated quality control (Section 5.5.2) of accelerometer data during walking.

Chapter 6 posed the problem of occupancy estimation using single PIR sensor. We developed a robust and economically valuable solution for this problem using iHMM trained with online iterative MAP method. It was shown that the accuracy of the system reduces only marginally when we using our MAP method compare to existing MCMC sampling methods. At the same time with our proposed approach we could segment incoming streams of data for up to 1.15 seconds, using only the computational power of a microcontroller board. The efficiency of MAP-iHMM allowed us to design a self-contained system that runs without interruptions. The prototype could also dynamically update efficient summaries of the iHMM, on board, adapting to new data, without the need to keep too large sets of raw data itself.

7.2 Future directions

BNP models have proven extremely useful for defining flexible probabilistic models that can adequately handle the uncertainty in modern datasets. In recent years, supervised pattern recognition methods have been on the rise outperforming humans in many sophisticated tasks. For example, deep learning neural networks are continuously being deployed in practice for well defined learning tasks such as object recognition from images (Hinton *et al.*, 2006; LeCun *et al.*, 2015; He *et al.*, 2016); reinforcement learning methods coupled with deep neural network classifiers has made a recent breakthrough by mastering the classical board game of Go (Silver *et al.*, 2016); random forests methods remain some of the most used pattern recognition techniques for classification and regression. At the same time high performance techniques like deep learning or random forest classifiers often trade away any intuitive interpretation of how the algorithms reach to a decision. This makes them very poor in handling uncertainty and particularly sensitive to model misspecification as recent work has hinted (Szegedy *et al.*, 2013; Nguyen *et al.*, 2015). For a wide range of data science problems we still need adequate and scalable ways to model uncertainty which can potentially come from advances in models and inference techniques for the transparent and flexible BNP latent variable models.

7.2.1 Unsupervised behaviour modeling

In Section 5.5.2 we demonstrated how MAP-iHMM can be used to automatically segment out some undesired biasing behaviours from streams of accelerometer data in order to produce better quality gait data and make better inference about users PD diagnosis and progression. Since MAP-iHMM is based on the simple iHMM with Gaussian emissions, it clusters separately only behaviours described with different Gaussian distributions. Similarly in Section 6.4 we used MAP-iHMM this time with Laplace emissions to segment out biasing motion patterns from streams of PIR sensor data. This step helped us then to focus only on the sensor data which was more descriptive of the occupancy count. In both cases we used the iHMM to model changing behavioural patterns of users or occupants in order to later focus only on the bahaviours most relevant for the problem at hand. However, using simple emission mechanism for the iHMM (Gaussian distribution or Laplace distribution) we do not take into account the dynamics of the different behaviours. This is why we struggle to segment behaviours which have very subtle differences, like for example different walking patterns.

Fox *et al.* (2009) have proposed more flexible extensions of the infinite HMM can overcome such issues and can be used to more accurately model accurately human motion data. BNP models such as the *nonparametric switching vector autoregressive models* and the *nonparametric switching linear dynamical systems* Fox *et al.* (2009) can potentially enable us to model a very rich set of user behaviours. Deriving efficient MAP methods for such more flexible extensions of the iHMM would enable us to extend the potential applications for the simpler MAP-iHMM method. For example, iterative MAP methods based on the nonparametric switching vector autoregressive model could allow us to automate most of the quality control labor involved in using PD smarthphone data. In the case of occupancy estimation, more flexible segmentation approach would improve our overall occupancy estimates and increase the responsiveness of the developed system to the changing environment.

7.2.2 Real time learning

In many applications it is beneficial to process data as soon as it arrives, on board of a resource constrained data collection device. For example, in Internet of Things applications small devices are constantly generating vast amounts of sensor data; this data typically has to be transferred to a cloud and any processing has to be done using external hardware. However, this data transfer delays the inference process, drains the battery of the IoT devices and furthermore can often involve transfer of a large amount of redundant data.

If we perform at least some of the processing of the data directly onto the embedded hardware, we can avoid the need to transfer and interpret redundant data, we could improve the functionality of the devices and at the same time extend their battery life. In order to discover sufficient structure from the streaming sensor data which can be used and transferred in a compact way we need flexible models which can adapt to the changing environment and self-learn Iyer & Ozer (2016). A promising candidate for modeling and representing IoT sensor data and enriching the IoT applications are BNP models. They have been designed for exactly the setup of forming self-adapting autonomous systems because they can grow in complexity without any re-training. Unfortunately the use of BNP models in IoT applications has been somewhat limited by the prohibitive demanding inference methods commonly used of inference in BNP models. In Chapter 6 we demonstrated that MAP methods can be a useful alternative which allows us to infer flexible models such as the iHMM directly on board of an energy constrained microcontroller board. While the occupancy estimation is interesting problem in its own right, we believe that the proposed modeling framework can be significantly extended in order to build self-aware more responsive smart environments and buildings.

7.2.3 Parallel iterative MAP methods

Whether an inference algorithm is ran online on streaming data or offline on large batches of data, modern big data problems require adequate utilization of the computational hardware available. The most common way to do so is through some form of parallel computation at either data and/or task level. Parallelism is a relatively old concept employed for many years, but in recent years it is re-gaining a lot of popularity from the data science community mainly due to the rapid increase in the capability of data acquisition.

As inference methods for BNP models are not embarrassingly parallel, some considerable effort is required in order to employ any significant parallelism for their scaling. Towards this end Lovell *et al.* (2012) and Williamson *et al.* (2013) have proposed parallel MCMC methods for inference in DPMM, Bratieres *et al.* (2010) has proposed distributed MCMC inference method for the infinite HMM and some considerable efforts have been made for deriving parallel MCMC inference in other popular BNP models. Nevertheless,, as Gal & Ghahramani (2014) points out the practical advantages of some of the current methods for parallel inference are quite limited. Hence parallel MCMC methods can also be quite slow even after employing parallelism and the existing parallelism schemes do not extend directly to scalable MAP methods.

In Section 5.4.2 we demonstrated one efficient way to employ parallelism for MAP inference in the infinite HMM, however in future it will be useful to derive related parallel MAP methods for inference in DPMM, HDP mixtures and other BNP models. A nice consequence of the MAP method proposed in Section 5.4.2 was that in addition to employing parallelism it allowed for inference in non-conjugate iHMM models.

Bibliography

Agarwal, Yuvraj, Balaji, Bharathan, Gupta, Rajesh, Lyles, Jacob, Wei, Michael, & Weng, Thomas. 2010. Occupancy-driven energy management for smart building automation. *Pages 1–6 of: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM.

Albert, James H, & Chib, Siddhartha. 1993. Bayes inference via Gibbs sampling of autoregressive time series subject to Markov mean and variance shifts. *Journal of Business & Economic Statistics*, **11**(1), 1–15.

Amari, Shun-Ichi. 1998. Natural gradient works efficiently in learning. *Neural computation*, **10**(2), 251–276.

Andrade, Ernesto L, Blunsden, Scott, & Fisher, Robert B. 2006. Hidden markov models for optical flow analysis in crowds. *Pages 460–463 of: 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE.

Antoniak, Charles E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics*, 1152–1174.

Arora, Siddharth, Venkataraman, Vinayak, Donohue, Sean, Biglan, Kevin M, Dorsey, Earl R, & Little, Max A. 2014. High accuracy discrimination of Parkinson's disease participants from healthy controls using smartphones. *Pages 3641–3644 of: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Baldi, Pierre, & Brunak, Søren. 2001. *Bioinformatics: the machine learning approach*. MIT press.

Banerjee, Arindam, Merugu, Srujana, Dhillon, Inderjit S, & Ghosh, Joydeep. 2005. Clustering with Bregman divergences. *Journal of machine learning research*, **6**(Oct), 1705–1749.

Baum, Leonard E, Petrie, Ted, Soules, George, & Weiss, Norman. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, **41**(1), 164–171.

Beal, Matthew J, Ghahramani, Zoubin, & Rasmussen, Carl E. 2002. The infinite hidden Markov model. *Pages 577–584 of: Advances in neural information processing systems*.

Berkhin, Pavel. 2006. *A survey of clustering data mining techniques*. Springer. Pages 25–71.

Bertoletti, Marco, Friel, Nial, & Rastelli, Riccardo. 2015. Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion. *Metron*, **73**(2), 177–199.

Besag, Julian. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 259–302.

Bischof, Horst, Leonardis, Aleš, & Selb, Alexander. 1999. MDL principle for robust vector quantisation. *Pattern Analysis & Applications*, **2**(1), 59–72.

Bishop, Christopher M. 1995. *Neural networks for pattern recognition*. Oxford university press.

Bishop, Christopher M. 2006. Pattern recognition. *Machine Learning*, **128**.

Bishop, Christopher M. 2013. Model-based machine learning. *Phil. Trans. R. Soc. A*, **371**(1984), 20120222.

Bitvai, Zsolt, & Cohn, Trevor. 2015. Non-linear text regression with a deep convolutional neural network. *Pages 180–185 of: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2.

Blackwell, David. 1947. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 105–110.

Blake, Catherine, & Merz, Christopher J. 1998. {UCI} Repository of machine learning databases.

Blei, David M, & Jordan, Michael I. 2006. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, **1**(1), 121–144.

Bousquet, Olivier, & Bottou, Léon. 2008. The tradeoffs of large scale learning. *Pages 161–168 of: Advances in neural information processing systems*.

Bratieres, Sébastien, Van Gael, Jurgen, Vlachos, Andreas, & Ghahramani, Zoubin. 2010. Scaling the iHMM: parallelization versus Hadoop. *Pages 1235–1240 of: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*. IEEE.

Brett, Megan R. 2012. Topic modeling: a basic introduction. *Journal of Digital Humanities*, **2**(1), 12–16.

Broderick, Tamara, Kulis, Brian, & Jordan, Michael I. 2013a. MAD-Bayes: MAP-based Asymptotic Derivations from Bayes. *Pages 226–234 of: ICML (3)*.

Broderick, Tamara, Boyd, Nicholas, Wibisono, Andre, Wilson, Ashia C, & Jordan, Michael I. 2013b. Streaming variational bayes. *Pages 1727–1735 of: Advances in Neural Information Processing Systems*.

Brooks, Rodney A. 1997. The intelligent room project. *Pages 271–278 of: Cognitive Technology, 1997. Humanizing the Information Age. Proceedings., Second International Conference on*. IEEE.

Bryant, Michael, & Sudderth, Erik B. 2012. Truly nonparametric online variational inference for hierarchical Dirichlet processes. *Pages 2699–2707 of: Advances in Neural Information Processing Systems*.

Carota, Cinzia, & Parmigiani, Giovanni. 2002. Semiparametric regression for count data. *Biometrika*, **89**(2), 265–281.

Chan, Antoni B, & Vasconcelos, Nuno. 2012. Counting people with low-level features and Bayesian regression. *Image Processing, IEEE Transactions on*, **21**(4), 2160–2177.

Chib, Siddhartha, & Greenberg, Edward. 1995. Understanding the metropolis-hastings algorithm. *The american statistician*, **49**(4), 327–335.

Chung, Pau-Choo, & Liu, Chin-De. 2008. A daily behavior enabled hidden Markov model for human behavior understanding. *Pattern Recognition*, **41**(5), 1572–1580.

Cifarelli, D, & Regazzini, E. 1978. *Problemi statistici non parametrici in condizioni di scambiabilita parziale e impiego di medie associative*. Report. Tech. rep., Quaderni Istituto Matematica Finanziaria dell Universita di Torino.

Colwell, Bob. 2013. End of Moore's Law.

Dahl, David B, *et al.* 2009. Modal clustering in a class of product partition models. *Bayesian Analysis*, **4**(2), 243–264.

Damlen, P, Wakefield, John, & Walker, Stephen. 1999. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**(2), 331–344.

Daumé III, Hal. 2007. Fast search for Dirichlet process mixture models. *Page 3245 of: AISTATS*, vol. 3244.

Dempster, Arthur P, Laird, Nan M, & Rubin, Donald B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Dodier, Robert H, Henze, Gregor P, Tiller, Dale K, & Guo, Xin. 2006. Building occupancy detection through sensor belief networks. *Energy and buildings*, **38**(9), 1033–1043.

Dunson, David B. 2006. Bayesian dynamic modeling of latent trait distributions. *Biostatistics*, **7**(4), 551–568.

Dunson, David B. 2010. Nonparametric Bayes applications to biostatistics. *Bayesian nonparametrics*, **28**, 223.

Durbin, Richard, Eddy, Sean R, Krogh, Anders, & Mitchison, Graeme. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

Edwards, Robert G., & Sokal, Alan D. 1988. Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Phys. Rev. D*, **38**(Sep), 2009–2012.

Erickson, Varick L, Lin, Yiqing, Kamthe, Ankur, Brahme, Rohini, Surana, Amit, Cerpa, Alberto E, Sohn, Michael D, & Narayanan, Satish. 2009. Energy efficient building environment control strategies using real-time occupancy measurements. *Pages 19–24 of: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM.

Fan, Jianqing. 1992. Design-adaptive nonparametric regression. *Journal of the American statistical Association*, **87**(420), 998–1004.

Fei-Fei, Li, & Perona, Pietro. 2005. A bayesian hierarchical model for learning natural scene categories. *Pages 524–531 of: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE.

Forster, Jürgen, & Warmuth, Manfred K. 2002. Relative expected instantaneous loss bounds. *Journal of Computer and System Sciences*, **64**(1), 76–102.

Foti, Nicholas, Xu, Jason, Laird, Dillon, & Fox, Emily. 2014. Stochastic variational inference for hidden Markov models. *Pages 3599–3607 of: Advances in Neural Information Processing Systems*.

Fox, Emily, Sudderth, Erik B, Jordan, Michael I, & Willsky, Alan S. 2009. Nonparametric Bayesian learning of switching linear dynamical systems. *Pages 457–464 of: Advances in Neural Information Processing Systems*.

Fox, Emily B, Sudderth, Erik B, Jordan, Michael I, & Willsky, Alan S. 2011. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, 1020–1056.

Gal, Yarin, & Ghahramani, Zoubin. 2014. Pitfalls in the use of Parallel Inference for the Dirichlet Process. *Pages 208–216 of: ICML*.

Gao, Debin, Reiter, Michael K, & Song, Dawn. 2006. Behavioral distance measurement using hidden markov models. *Pages 19–40 of: International Workshop on Recent Advances in Intrusion Detection*. Springer.

Gao, Hong, Bryc, Katarzyna, & Bustamante, Carlos D. 2011. On identifying the optimal number of population clusters via the deviance information criterion. *PloS one*, **6**(6), e21014.

Gasparoli, E, Delibori, D, Polesello, G, Santelli, L, Ermani, M, Battistin, L, & Bracco, F. 2002. Clinical predictors in Parkinson's disease. *Neurological Sciences*, **23**(2), s77–s78.

Geman, Stuart, & Geman, Donald. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 721–741.

Gershman, Samuel J, & Blei, David M. 2012. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, **56**(1), 1–12.

Gilks, Walter R, & Wild, Pascal. 1992. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 337–348.

Hashimoto, Kazuhiko, Morinaka, Katsuya, Yoshiike, Nobuyuki, Kawaguchi, Chihiro, & Matsueda, Satoshi. 1997. People count system using multi-sensing application. *Pages 1291–1294 of: Solid State Sensors and Actuators, 1997. TRANSDUCERS'97 Chicago., 1997 International Conference on*, vol. 2. IEEE.

Hastings, W Keith. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2016. Deep residual learning for image recognition. *Pages 770–778 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hinton, Geoffrey E, Osindero, Simon, & Teh, Yee-Whye. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, **18**(7), 1527–1554.

Hjort, Nils Lid, Holmes, Chris, Müller, Peter, & Walker, Stephen G. 2010. *Bayesian nonparametrics*. Vol. 28. Cambridge University Press.

Hoehn, Margaret M, Yahr, Melvin D, *et al.* 1998. Parkinsonism: onset, progression, and mortality. *Neurology*, **50**(2), 318–318.

Hughes, Michael C, & Sudderth, Erik. 2013. Memoized online variational inference for Dirichlet process mixture models. *Pages 1133–1141 of: Advances in Neural Information Processing Systems*.

Hughes, Michael C, Fox, Emily, & Sudderth, Erik B. 2012. Effective split-merge monte carlo methods for nonparametric models of sequential data. *Pages 1295–1303 of: Advances in Neural Information Processing Systems*.

Hughes, Michael C, Kim, Dae Il, & Sudderth, Erik B. 2015a. Reliable and Scalable Variational Inference for the Hierarchical Dirichlet Process. *In: AISTATS*.

Hughes, Michael C, Stephenson, William T, & Sudderth, Erik. 2015b. Scalable Adaptation of State Complexity for Nonparametric Hidden Markov Models. *Pages 1198–1206 of: Advances in Neural Information Processing Systems*.

Iyer, Ravi, & Ozer, Emre. 2016. Visual IoT: Architectural Challenges and Opportunities; Toward a Self-Learning and Energy-Neutral IoT. *IEEE Micro*, **36**(6), 45–49.

Jelinek, Frederick. 1997. *Statistical methods for speech recognition*. MIT press.

Jiang, Ke, Kulis, Brian, & Jordan, Michael I. 2013. Small-variance asymptotics for exponential family Dirichlet process mixture models. *Pages 3158–3166 of: Advances in Neural Information Processing Systems*.

Johnson, Matthew, & Willsky, Alan S. 2014. Stochastic Variational Inference for Bayesian Time Series Models. *Pages 1854–1862 of: ICML*.

Kang, Le, Ye, Peng, Li, Yi, & Doermann, David. 2014. Convolutional neural networks for no-reference image quality assessment. *Pages 1733–1740 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Khan, Aftab, Nicholson, James, Mellor, Sebastian, Jackson, Daniel, Ladha, Karim, Ladha, Cassim, Hand, Jon, Clarke, Joseph, Olivier, Patrick, & Plötz, Thomas. 2014. Occupancy monitoring using environmental and context sensors and a hierarchical analysis framework. *Pages 90–99 of: BuildSys@ SenSys*.

Kientz, Julie A, Patel, Shwetak N, Jones, Brian, Price, ED, Mynatt, Elizabeth D, & Abowd, Gregory D. 2008. The georgia tech aware home. *Pages 3675–3680 of: CHI'08 extended abstracts on Human factors in computing systems*. ACM.

Kim, Seyoung, & Smyth, Padhraic. 2006. Hierarchical Dirichlet processes with random effects. *Pages 697–704 of: Advances in Neural Information Processing Systems*.

Kish, Laszlo B. 2002. End of Moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, **305**(3), 144–149.

Kleinman, Ken P, & Ibrahim, Joseph G. 1998. A semiparametric Bayesian approach to the random effects model. *Biometrics*, 921–938.

Kolmogorov, Andrei Nikolaevich. 1950. Unbiased estimates. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, **14**(4), 303–326.

Krogh, Anders, Brown, Michael, Mian, I Saira, Sjölander, Kimmen, & Haussler, David. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, **235**(5), 1501–1531.

Kulis, Brian, & Jordan, Michael I. 2011. Revisiting k-means: New algorithms via Bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*.

Kurlan, Roger, & Murphy, Diane. 2007. Parkinson's disease data and organizing center. *Movement disorders: official journal of the Movement Disorder Society*, **22**(6), 904.

Lam, Khee Poh, Höynck, Michael, Dong, Bing, Andrews, Burton, Chiou, Yun-Shang, Zhang, Rui, Benitez, Diego, & Choi, Joonho. 2009. Occupancy detection through an extensive environmental sensor network in an open-plan office building. *IBPSA Building Simulation*, **145**, 1452–1459.

LeCun, Yann, Bengio, Yoshua, & Hinton, Geoffrey. 2015. Deep learning. *Nature*, **521**(7553), 436–444.

Lempitsky, Victor, & Zisserman, Andrew. 2010. Learning to count objects in images. *Pages 1324–1332 of: Advances in Neural Information Processing Systems.*

Lewis, SJG, Foltynie, T, Blackwell, AD, Robbins, TW, Owen, AM, & Barker, RA. 2005. Heterogeneity of Parkinson's disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, **76**(3), 343–348.

Liang, Percy, Petrov, Slav, Jordan, Michael I, & Klein, Dan. 2007. The Infinite PCFG Using Hierarchical Dirichlet Processes. *Pages 688–697 of: EMNLP-CoNLL.*

Little, Max A, & Jones, Nick S. 2011. Generalized methods and solvers for noise removal from piecewise constant signals. I. Background theory. *Pages 3088–3114 of: Proc. R. Soc. A*, vol. 467. The Royal Society.

Liu, Jun S, Wong, Wing Hung, & Kong, Augustine. 1994. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, **81**(1), 27–40.

Liu, Ping, Feng, Tao, Wang, Yong-jun, Zhang, Xuan, & Chen, Biao. 2011. Clinical heterogeneity in patients with early-stage Parkinson's disease: a cluster analysis. *Journal of Zhejiang University Science B*, **12**(9), 694–703.

Lloyd, Stuart. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, **28**(2), 129–137.

Lovell, Dan, Adams, Ryan P, & Mansingka, VK. 2012. Parallel markov chain monte carlo for dirichlet process mixtures. *In: Workshop on Big Learning, NIPS.*

Manning, Christopher D, & Schütze, Hinrich. 1999. *Foundations of statistical natural language processing.* Vol. 999. MIT Press.

Mannini, Andrea, & Sabatini, Angelo Maria. 2012. Gait phase detection and discrimination between walking and jogging activities using hidden Markov models applied to foot motion data from a gyroscope. *Gait and posture*, **36**(4), 657–661.

Martin, James H, & Jurafsky, Daniel. 2000. Speech and language processing. *International Edition*, **710**.

Meyn, Sean, Surana, Amit, Lin, Yiqing, Oggianu, Stella M, Narayanan, Satish, & Frewen, Thomas A. 2009. A sensor-utility-network method for estimation of occupancy in buildings. *Pages 1494–1500 of: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE.

Miller, Jeffrey W, & Harrison, Matthew T. 2013. A simple example of Dirichlet process mixtuure inconsistency for the number of components. *Pages 199–206 of: Advances in neural information processing systems.*

Molenberghs, Geert, Beunckens, Caroline, Sotto, Cristina, & Kenward, Michael G. 2008. Every missingness not at random model has a missingness at random counterpart with equal fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**(2), 371–388.

Morris, Carl N. 1983. Parametric empirical Bayes inference: theory and applications. *Journal of the American Statistical Association*, **78**(381), 47–55.

Muliere, Pietro, & Petrone, Sonia. 1993. A Bayesian predictive approach to sequential search for an optimal dose: parametric and nonparametric models. *Journal of the Italian Statistical Society*, **2**(3), 349–364.

Müller, Peter, Quintana, Fernando, & Rosner, Gary. 2004. A method for combining inference across related nonparametric Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **66**(3), 735–749.

Nag, R, Wong, K, & Fallside, Frank. 1986. Script recognition using hidden Markov models. *Pages 2071–2074 of: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, vol. 11. IEEE.

Neal, Radford M. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, **9**(2), 249–265.

Neal, Radford M. 2003. Slice sampling. *Annals of statistics*, 705–741.

Netuveli, Gopalakrishnan, Wiggins, Richard D, Hildon, Zoe, Montgomery, Scott M, & Blane, David. 2006. Quality of life at older ages: evidence from the English longitudinal study of aging (wave 1). *Journal of Epidemiology and Community Health*, **60**(4), 357–363.

Nguyen, Anh, Yosinski, Jason, & Clune, Jeff. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Pages 427–436 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Paisley, John, Wang, Chong, Blei, David M, & Jordan, Michael I. 2012. Nested hierarchical Dirichlet processes. *arXiv preprint arXiv:1210.6738*.

Parisi, Giorgio. 1988. *Statistical field theory*. Addison-Wesley.

Pathak, Deepak, Krahenbuhl, Philipp, Yu, Stella X, & Darrell, Trevor. 2015. Constrained Structured Regression with Convolutional Neural Networks. *arXiv preprint arXiv:1511.07497*.

Pelleg, Dan, Moore, Andrew W, *et al.* 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *In: ICML*, vol. 1.

Pitman, Jim, & Yor, Marc. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 855–900.

Rabiner, Lawrence, & Juang, B. 1986. An introduction to hidden Markov models. *ieee assp magazine*, **3**(1), 4–16.

Rabiner, Lawrence, & Juang, Biing-Hwang. 1993. Fundamentals of speech recognition.

Raftery, Adrian E, & Lewis, Steven M. 1992. [Practical Markov Chain Monte Carlo]: comment: one long run with diagnostics: implementation strategies for Markov Chain Monte Carlo. *Statistical science*, **7**(4), 493–497.

Rasmussen, Carl Edward. 1999. The infinite Gaussian mixture model. *Pages 554–560 of: NIPS*, vol. 12.

Raykov, Yordan P, Boukouvalas, Alexis, & Little, Max A. 2014. Simple approximate MAP Inference for Dirichlet processes. *arXiv preprint arXiv:1411.0939*.

Raykov, Yordan P, Boukouvalas, Alexis, & Little, Max A. 2015a. Iterative collapsed MAP inference for Bayesian nonparametrics.

Raykov, Yordan P, Boukouvalas, Alexis, & Little, Max A. 2015b. MAP for Exponential Family Dirichlet Process Mixture Models.

Raykov, Yordan P, ARM, Cambridge, Ozer, Emre, Dasika, Ganesh, & Little, Max A. 2016a. Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction. *Pages 1016–1027 of: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM.

Raykov, Yordan P, Boukouvalas, Alexis, Little, Max A, *et al.* 2016b. Simple approximate MAP inference for Dirichlet processes mixtures. *Electronic Journal of Statistics*, **10**(2), 3548–3578.

Raykov, Yordan P, Boukouvalas, Alexis, Baig, Fahd, & Little, Max A. 2016c. What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. *PloS one*, **11**(9), e0162259.

Reijnders, JSAM, Ehrt, U, Lousberg, R, Aarsland, D, & Leentjens, AFG. 2009. The association between motor subtypes and psychopathology in Parkinson's disease. *Parkinsonism & related disorders*, **15**(5), 379–382.

Ren, Lu, Dunson, David B, & Carin, Lawrence. 2008. The dynamic hierarchical Dirichlet process. *Pages 824–831 of: Proceedings of the 25th international conference on Machine learning*. ACM.

Robert, Christian P, Celeux, Gilles, & Diebolt, Jean. 1993. Bayesian estimation of hidden Markov chains: A stochastic implementation. *Statistics & Probability Letters*, **16**(1), 77–83.

Roweis, Sam. 1998. EM algorithms for PCA and SPCA. *Advances in neural information processing systems*, 626–632.

Roychowdhury, Anirban, Jiang, Ke, & Kulis, Brian. 2013. Small-variance asymptotics for hidden Markov models. *Pages 2103–2111 of: Advances in Neural Information Processing Systems*.

Rubin, Donald B. 1976. Inference and missing data. *Biometrika*, **63**(3), 581–592.

Salakhutdinov, Ruslan, Tenenbaum, Joshua B, & Torralba, Antonio. 2013. Learning with hierarchical-deep models. *IEEE transactions on pattern analysis and machine intelligence*, **35**(8), 1958–1971.

Sammut, Claude, & Webb, Geoffrey I. 2011. *Encyclopedia of machine learning*. Springer Science & Business Media.

Schaller, Robert R. 1997. Moore's law: past, present and future. *IEEE spectrum*, **34**(6), 52–59.

Scott, Steven L. 2002. Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *Journal of the American Statistical Association*, **97**(457), 337–351.

Scott, Steven L. 2011. Bayesian methods for hidden Markov models. *Journal of the American Statistical Association*.

Sethuraman, Jayaram. 1994. A constructive definition of Dirichlet priors. *Statistica sinica*, 639–650.

Silver, David, Huang, Aja, Maddison, Chris J, Guez, Arthur, Sifre, Laurent, Van Den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Panneershelvam, Veda, Lanctot, Marc, *et al.* 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, **529**(7587), 484–489.

Smola, Alex, & Vapnik, Vladimir. 1997. Support vector regression machines. *Advances in neural information processing systems*, **9**, 155–161.

Snijders, Antoine M, Nowak, Norma, Segraves, Richard, Blackwood, Stephanie, Brown, Nils, Conroy, Jeffrey, Hamilton, Greg, Hindle, Anna Katherine, Huey, Bing, Kimura, Karen, *et al.* 2001. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature genetics*, **29**(3), 263–264.

Snoek, Jasper, Larochelle, Hugo, & Adams, Ryan P. 2012. Practical bayesian optimization of machine learning algorithms. *Pages 2951–2959 of: Advances in neural information processing systems.*

Spearman, Charles. 1904. " General Intelligence," objectively determined and measured. *The American Journal of Psychology*, **15**(2), 201–292.

Sudderth, Erik B. 2006. *Graphical models for visual object recognition and tracking.* Ph.D. thesis, Citeseer.

Sung, K-K, & Poggio, Tomaso. 1998. Example-based learning for view-based human face detection. *IEEE Transactions on pattern analysis and machine intelligence*, **20**(1), 39–51.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, & Fergus, Rob. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199.*

Tank, Alex, Foti, Nicholas J, & Fox, Emily B. 2015. Streaming Variational Inference for Bayesian Nonparametric Mixture Models. *In: AISTATS.*

Teh, Yee W, Kurihara, Kenichi, & Welling, Max. 2007. Collapsed variational inference for HDP. *Pages 1481–1488 of: Advances in neural information processing systems.*

Teh, Yee Whye, Jordan, Michael I, Beal, Matthew J, & Blei, David M. 2006. Hierarchical dirichlet processes. *Journal of the american statistical association.*

Tipping, Michael E, & Bishop, Christopher M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**(3), 611–622.

Toreyin, B. Ugur, Soyer, E. Birey, Urfalioglu, O., & Cetin, A. Enis. 2008. Flame detection system based on wavelet analysis of PIR sensor signals with an HMM decision mechanism. *Pages 1–5 of: Signal Processing Conference, 2008 16th European.*

Van Gael, Jurgen. 2012. *Bayesian Nonparametric Hidden Markov Models.* Ph.D. thesis, Citeseer.

Van Gael, Jurgen, Saatci, Yunus, Teh, Yee Whye, & Ghahramani, Zoubin. 2008. Beam sampling for the infinite hidden Markov model. *Pages 1088–1095 of: Proceedings of the 25th international conference on Machine learning.* ACM.

Van Oosterhout, Tim, Bakkes, Sander, & Kröse, Ben JA. 2011. Head Detection in Stereo Data for People Counting and Segmentation. *Pages 620–625 of: VISAPP.*

van Rooden, Stephanie M, Heiser, Willem J, Kok, Joost N, Verbaan, Dagmar, van Hilten, Jacobus J, & Marinus, Johan. 2010. The identification of Parkinson's disease subtypes using cluster analysis: a systematic review. *Movement Disorders*, **25**(8), 969–978.

Viterbi, Andrew. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, **13**(2), 260–269.

Wahl, Florian, Milenkovic, Milo, & Amft, Oliver. 2012. A distributed PIR-based approach for estimating people count in office environments. *Pages 640–647 of: Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*. IEEE.

Walker, Stephen G. 2007. Sampling the Dirichlet mixture model with slices. *Communications in Statistics-Simulation and Computation*, **36**(1), 45–54.

Wang, Chong, Paisley, John William, & Blei, David M. 2011. Online Variational Inference for the Hierarchical Dirichlet Process. *Page 4 of: AISTATS*, vol. 2.

Wang, Lianming, & Dunson, David B. 2011. Fast Bayesian inference in Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, **20**(1), 196–216.

Wang, Liming, & Wang, Xiaodong. 2013. Hierarchical Dirichlet process model for gene expression clustering. *EURASIP Journal on Bioinformatics and Systems Biology*, **2013**(1), 1.

Welling, Max, & Kurihara, Kenichi. 2006. Bayesian K-Means as a" Maximization-Expectation" Algorithm. *Pages 474–478 of: SDM*. SIAM.

Williams, Christopher KI. 1998. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. *Pages 599–621 of: Learning in graphical models*. Springer.

Williamson, Sinead, Dubey, Avinava, & Xing, Eric P. 2013. Parallel Markov Chain Monte Carlo for Nonparametric Mixture Models. *Pages 98–106 of: ICML (1)*.

Wright, Stephen J. 2015. Coordinate descent algorithms. *Mathematical Programming*, **151**(1), 3–34.

Yang, Danny B, Gonzalez-Banos, Hector H, & Guibas, Leonidas J. 2003. Counting people in crowds with a real-time network of simple image sensors. *Pages 122–129 of: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE.

Yang, Hui-Jun, Kim, Young Eun, Yun, Ji Young, Kim, Han-Joon, & Jeon, Beom Seok. 2014. Identifying the clusters within nonmotor manifestations in early Parkinson's disease by using unsupervised cluster analysis. *PloS one*, **9**(3), e91906.

Yun, Jaeseok, & Lee, Sang-Shin. 2014. Human Movement Detection and Identification Using Pyroelectric Infrared Sensors. *Sensors*, **14**(5), 8057.

Zappi, P., Farella, E., & Benini, L. 2010. Tracking Motion Direction and Distance With Pyroelectric IR Sensors. *IEEE Sensors Journal*, **10**(9), 1486–1494.

Zappi, Piero, Farella, Elisabetta, & Benini, Luca. 2007. Enhancing the spatial resolution of presence detection in a PIR based wireless surveillance network. *Pages 295–300 of: Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE.

Appendix A

Hyper parameters updates for exponential family conjugate pairs

In the generalized MAP-DP algorithm (Algorithm (3.8.1)), the computation of the variables $d_{i,k}$ and $d_{i,K+1}$ (Algorithm(3.8.1) lines 8,9) requires the collapsed prior predictive distribution $f\left(x|\theta_0\right)$, and also the collapsed posterior predictive distribution $f\left(x|\theta_k^{-i}\right)$. This predictive distribution requires the updated cluster posterior hyper parameters θ_k^{-i} (algorithm line 7). These updates depend upon the distribution, and the data type, of each data point x_i. When the distribution is from the *exponential family*, the prior distribution over the parameters can be chosen to be *conjugate*: the prior over the parameters of the data distribution and the posterior have the same form of distribution. This simplifies the hyper parameter updates, and, furthermore, the form of the prior and posterior predictive distributions is the same and is available in closed form. The table below lists some possible data types and distributions, their conjugate prior/posterior distribution, the names given to the hyper parameters and the corresponding name of the predictive distributions. We discuss each case in more detail in the subsequent sections.

Distribution of data x_i	Data type	Conjugate prior/posterior	Hyper parameters θ	Predictive distribution $f\left(x\|\theta\right)$
Spherical normal (known variance)	$x \in \mathbb{R}^D$	Spherical normal	$\left(\mu, \sigma^2\right)$	Spherical normal
Multivariate normal (known covariance)	$x \in \mathbb{R}^D$	Multivariate normal	(μ, Σ)	Multivariate normal
Multivariate normal	$x \in \mathbb{R}^D$	Normal-Wishart	(m, c, B, a)	Multivariate Student-t
Exponential	$x \in \mathbb{R},\, x \geq 0$	Gamma	(α, β)	Lomax
Categorical	$x \in \{1, 2, \ldots D\}$	Dirichlet	$(\alpha_1, \ldots, \alpha_D)$	Dirichlet-multinomial
Binomial	$x \in \{0, 1, \ldots n\}$	Beta	(α, β)	Beta-binomial
Poisson	$x \in \mathbb{Z},\, x \geq 0$	Gamma	(α, β)	Negative-binomial
Geometric	$x \in \mathbb{Z},\, x \geq 0$	Beta	(α, β)	Ratio of beta functions

Spherical normal data with known variance

This is the variant of MAP-DP described in Algorithm (3.8.1). When each data point $x \in \mathbb{R}^D$ is assumed to be spherical Gaussian with known variance $\hat{\sigma}^2$ shared across dimensions, the conjugate prior distribution of the Gaussian mean vector parameter $\mu \in \mathbb{R}^D$ is also spherical normal with hyper parameters $\theta_0 = \left(\mu_0, \sigma_0^2\right)$. Then the posterior distribution for each cluster is also spherical normal with hyper parameters $\theta_k^{-i} = \left(\mu_k^{-i}, \sigma_k^{-i}\right)$. The hyper parameter updates (Algorithm (3.8.1), line 7) for each cluster are:

$$\begin{aligned} \sigma_k^{-i} &= \left(\frac{1}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2} N_k^{-i}\right)^{-1} \\ \mu_k^{-i} &= \sigma_k^{-i} \left(\frac{\mu_0}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2} \sum_{j:z_j=k, j\neq i} x_j\right) \end{aligned} \tag{A.1}$$

The predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are D-dimensional spherical normal distributions, whose negative logs are:

$$-\ln f\left(x|\theta\right) = \frac{1}{2\left(\sigma^2 + \hat{\sigma}^2\right)} \left\|x - \mu\right\|_2^2 + \frac{D}{2} \ln\left(\sigma^2 + \hat{\sigma}^2\right) + \frac{D}{2} \ln\left(2\pi\right) \tag{A.2}$$

Note that since the normalization term $\frac{D}{2} \ln\left(2\pi\right)$ is common to both predictive distributions, it can be omitted when computing $d_{i,k}$ and $d_{i,K+1}$ in the algorithm.

Multivariate normal data with known covariance

For data points $x \in \mathbb{R}^D$ assumed to be multivariate Gaussian with known covariance matrix $\hat{\Sigma}$, the conjugate prior distribution of the Gaussian mean vector parameter is also multivariate normal with hyper parameters $\theta_0 = \left(\mu_0, \Sigma_0\right)$. The posterior distribution for each cluster is also multivariate normal with hyper parameters $\theta_k^{-i} = \left(\mu_k^{-i}, \Sigma_k^{-i}\right)$. The hyper parameter updates are:

$$\begin{aligned} \Sigma_k^{-i} &= \left(\Sigma_0^{-1} + \hat{\Sigma}^{-1} N_k^{-i}\right)^{-1} \\ \mu_k^{-i} &= \Sigma_k^{-i} \left(\Sigma_0^{-1}\mu_0 + \hat{\Sigma}^{-1} \sum_{j:z_j=k, j\neq i} x_j\right) \end{aligned} \tag{A.3}$$

The predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are D-dimensional normal distributions, whose negative logs are:

$$-\ln f\left(x|\theta\right) = \tfrac{1}{2}\left(x - \mu\right)^T \left(\Sigma + \hat{\Sigma}\right)^{-1} \left(x - \mu\right) + \frac{D}{2} \ln\left|\Sigma + \hat{\Sigma}\right| + \frac{D}{2} \ln\left(2\pi\right) \tag{A.4}$$

Since the normalization term $\frac{D}{2} \ln\left(2\pi\right)$ is common to both predictive distributions, it can be omitted when computing $d_{i,k}$ and $d_{i,K+1}$ in the algorithm.

Multivariate Gaussian data

When each data point $x \in \mathbb{R}^D$ is assumed to be multivariate Gaussian with unknown mean vector and covariance matrix, the conjugate prior distribution of the Gaussian parameters is Normal-Wishart, with hyper parameters $\theta_0 = (m_0, c_0, B_0, a_0)$. Then, the posterior distribution for each cluster is also Normal-Wishart, with hyper parameters $\theta_k^{-i} = \left(m_k^{-i}, c_k^{-i}, B_k^{-i}, a_k^{-i}\right)$. These are updated for each cluster according to:

$$\begin{aligned} m_k^{-i} =& \frac{c_0 m_0 + N_k^{-i} \bar{x}_k^{-i}}{c_0 + N_k^{-i}} \\ c_k^{-i} =& c_0 + N_k^{-i} \\ B_k^{-i} =& \left(B_0^{-1} + S_k^{-i} + \frac{c_0 N_k^{-i}}{c_0 + N_k^{-i}} \left(\bar{x}_k^{-i} - m_0\right) \left(\bar{x}_k^{-i} - m_0\right)^T \right)^{-1} \\ a_k^{-i} =& a_0 + N_k^{-i} \end{aligned} \tag{A.5}$$

where:

$$\begin{aligned} \bar{x}_k^{-i} &= \frac{1}{N_k^{-i}} \sum_{j:z_j=k, j\neq i} x_j \\ S_k^{-i} &= \sum_{j:z_j=k, j\neq i} \left(x_i - \bar{x}_k^{-i}\right) \left(x_i - \bar{x}_k^{-i}\right)^T \end{aligned} \tag{A.6}$$

The predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are D-dimensional multivariate Student-t distributions, whose negative log, written in terms of the parameters (μ, Λ, ν) is:

$$\begin{aligned} -\ln f\left(x|\theta\right) = & \\ = & \frac{\nu + D}{2} \ln\left[1 + \nu^{-1} \left(x - \mu\right)^T \Lambda \left(x - \mu\right)\right] - \frac{1}{2} \ln|\Lambda| + \ln\Gamma\left(\frac{\nu}{2}\right) + \frac{D}{2} \ln\left(\nu\pi\right) - \ln\Gamma\left(\frac{\nu + D}{2}\right) \end{aligned} \tag{A.7}$$

where the Student-t parameters (μ, Λ, ν) are given in terms of the Normal-Wishart parameters $\mu = m$, $\nu = a - D + 1$ and $\Lambda = \frac{c\nu}{c+1} B$. We note that fast incremental updates of all these parameters are possible when including and then removing a single data point from a cluster, see (Raykov *et al.*, 2014) for further details.

Exponential data

Given data points $x \in \mathbb{R}$, $x \geq 0$ assumed to be exponentially-distributed, the conjugate prior over the exponential rate parameter is the gamma distribution. This gamma distribution has hyper parameters $\theta_0 = (\alpha, \beta)$ (shape, rate). So, the posterior probability of the rate parameter is also gamma, and the cluster hyper parameter $\theta_k^{-i} = \left(\alpha_k^{-i}, \beta_k^{-i}\right)$ are updated using:

$$\begin{aligned} \alpha_k^{-i} &= \alpha_0 + \sum_{j:z_j=k, j\neq i} x_j \\ \beta_k^{-i} &= \beta_0 + N_k^{-i} \end{aligned} \tag{A.8}$$

The predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are the so-called *Lomax* distribution, with negative

log:

$$-\ln f\left(x|\theta\right) = -\ln\alpha - \alpha\ln\beta + \left(\alpha+1\right)\ln\left(x+\beta\right) \tag{A.9}$$

Categorical data

For categorical data which can take on one of $D > 1$ possible values, $x \in \{1, 2, \ldots D\}$, the conjugate prior over the D outcome probability parameters of this distribution are Dirichlet distributed. This Dirichlet distribution has hyper parameters $\theta_0 = (\alpha_{0,1}, \ldots, \alpha_{0,D})$. So, the posterior outcome probability parameters for each cluster are also Dirichlet, and for each cluster the D entries in the cluster hyper parameter $\theta_k^{-i} = \alpha_k^{-i}$ are updated using:

$$\alpha_{k,d}^{-i} = \alpha_{0,d} + \sum_{j:z_j=k, j\neq i} \delta\left(x_j, d\right) \text{ for } d = 1, \ldots, D \tag{A.10}$$

where $\delta\left(x,y\right) = 1$ if $x = y$ and 0 otherwise. The predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are special cases of the Dirichlet-multinomial distribution, with negative log:

$$-\ln f\left(x|\theta\right) = -\ln\alpha_x + \ln\sum_{d=1}^{D}\alpha_d \tag{A.11}$$

Binomial data

In the case of binomial data where the data can take on $x \in \{0, 1, \ldots n\}$ for $n > 0$, the conjugate prior over the binomial success probability parameter is beta distributed, with hyper parameters $\theta_0 = (\alpha_0, \beta_0)$. By conjugacy, the posterior cluster parameters are also beta distributed with hyper parameters $\theta_k^{-i} = \left(\alpha_k^{-i}, \beta_k^{-i}\right)$, and are updated according to:

$$\begin{aligned} \alpha_k^{-i} &= \alpha_0 + \sum_{j:z_j=k, j\neq i} x_j \\ \beta_k^{-i} &= \beta_0 + N_k^{-i} n - \sum_{j:z_j=k, j\neq i} x_j \end{aligned} \tag{A.12}$$

For such binomial data, the predictive distributions $f\left(x|\theta_0\right)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are beta-binomial, with negative log:

$$-\ln f\left(x|\theta\right) = -\ln\left(\begin{array}{c} n \\ x \end{array}\right) - \ln B\left(x+\alpha, n-x+\beta\right) + \ln B\left(\alpha,\beta\right) \tag{A.13}$$

where $B\left(\cdot,\cdot\right)$ is the beta function:

$$B\left(\alpha,\beta\right) = \frac{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)}{\Gamma\left(\alpha+\beta\right)} \tag{A.14}$$

Poisson data

For positive integer Poisson count data $x \in \mathbb{Z}$, $x \geq 0$, the conjugate prior over the single rate parameter is the gamma distribution with hyper parameters $\theta_0 = (\alpha_0, \beta_0)$ (shape and rate, respectively). The posterior cluster parameters are similarly gamma distributed with hyper parameters $\theta_k^{-i} = \left(\alpha_k^{-i}, \beta_k^{-i}\right)$. The updates

for these hyper parameters are:

$$\begin{aligned} \alpha_k^{-i} &= \alpha_0 + \sum_{j:z_j=k, j\neq i} x_j \\ \beta_k^{-i} &= \beta_0 + N_k^{-i} \end{aligned} \tag{A.15}$$

For Poisson count data, the predictive distributions $f(x|\theta_0)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ are negative binomial distributed with negative log:

$$-\ln f(x|\theta) = -\ln \binom{\alpha+\beta-1}{\beta} - \alpha \ln(1-x) - \beta \ln x \tag{A.16}$$

Geometric data

In the case of positive integer data $x \in \mathbb{Z}$, $x \geq 0$ which is assumed to be geometrically-distributed, the conjugate prior over the single success probability parameter is the beta distribution with hyper parameters $\theta_0 = (\alpha_0, \beta_0)$. The posterior cluster parameters are similarly beta distributed with hyper parameters $\theta_k^{-i} = \left(\alpha_k^{-i}, \beta_k^{-i}\right)$. The updates for these hyper parameters are:

$$\begin{aligned} \alpha_k^{-i} &= \alpha_0 + N_k^{-i} \\ \beta_k^{-i} &= \beta_0 + \sum_{j:z_j=k, j\neq i} x_j \end{aligned} \tag{A.17}$$

For geometric data, the predictive distributions $f(x|\theta_0)$ and $f\left(x\left|\theta_k^{-i}\right.\right)$ have negative log:

$$-\ln f(x|\theta) = -\ln B(\alpha+1, \beta+x) + \ln B(\alpha, \beta) \tag{A.18}$$

where $B(\cdot,\cdot)$ is the beta function described above.

Appendix B

Implementation practicalities

As with all algorithms, implementation details can matter in practice. We discuss a few observations here:

- *Empty clusters.* In MAP-DP, as with K-means, it is always possible that a cluster ceases to have any data points assigned to it. In that case, since $N_k^{-i} = 0$, then it will be impossible in future iterations for data points to be assigned to that cluster label. So, it is reasonable to drop that label and re-assign the remaining non-empty clusters because the additional empty clusters are merely a wasted computational overhead. The MAP-DP algorithm (Algorithm (3.4)) can be readily modified to do this; the most sensible place to do this is immediately after lines 14 or 17.

- *Dominating reinforcement on initialization.* Collapsing out the cluster parameters causes the cluster geometry to be very robust, for example, largely insensitive to outliers. However, there is an unwanted side-effect of this robustness: because MAP-DP (Algorithm (3.8.1)) is initialized with one single large cluster, the reinforcement (rich-get-richer) effect of the DP can dominate over the geometry to cause MAP-DP to become trapped in the undesirable configuration where no new clusters can be generated. (Note that this is a problem for Gibbs sampling as well, but in theory at least, Gibbs can escape local minima after sufficient iterations, whereas MAP-DP cannot). Overcoming this reinforcement requires a prior count N_0 on the order of the magnitude of N, but this would usually create many spurious small clusters. To avoid this side-effect, a practical solution removes the reinforcement effect due to this particular initialization scheme by inserting $N_1^{-i} = 1$ in between lines 8 and 9 (Algorithm (3.4)), only on the first iteration.

- *Numerical computation of negative log likelihood.* Computing the NLL (Algorithm (3.4) line 17) requires evaluating $\ln \Gamma(N_k)$ terms which are difficult to estimate with high precision for large values of N_k. As a result the NLL can develop small numerical errors which can cause the NLL to increase slightly over iterations. A simple practical fix is to replace the convergence test with absolute values, i.e. $|E_{\text{old}} - E_{\text{new}}| < \epsilon$ in line 18.

B.1 Randomized restarts

As MAP-DP is a completely deterministic algorithm, if applied to the same data set with the same choice of input parameters, it will always produce the same clustering result. However, since the algorithm is not guaranteed to find the global maximum of the likelihood (in Equation (3.33)), it is important to attempt to restart the algorithm from different initial conditions to gain confidence that the MAP-DP clustering

solution is a good one. Since there are no random quantities at the start of the MAP-DP algorithm, one viable approach is to perform a random permutation of the order in which the data points are visited by the algorithm. The quantity E (the negative log of the expression in Equation (3.33)) at convergence can be compared across many random permutations of the ordering of the data, and the clustering partition with the lowest E chosen as the best estimate.

B.2 Obtaining cluster centroids

As explained in the introduction, MAP-DP does not explicitly compute estimates of the cluster centroids, but this is easy to do after convergence if required. The cluster posterior hyper parameters θ_k can be estimated using the appropriate Bayesian updating formulae for each data type, given in Appendix A. For example, for spherical normal data with known variance:

$$\begin{aligned} \sigma_k &= \left(\frac{1}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2} N_k\right)^{-1} \\ \mu_k &= \sigma_k \left(\frac{\mu_0}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2} \sum_{i:z_i=k} x_i\right) \end{aligned} \tag{B.1}$$

Using these parameters, useful properties of the posterior predictive distribution $f(x|\theta_k)$ can be computed, for example, in the case of spherical normal data, the posterior predictive distribution is itself normal, with mode μ_k. Indeed, this quantity plays an analogous role to the cluster means estimated using K-means.

Appendix C

Out-of-sample predictions

To make out-of-sample predictions we suggest two approaches to compute the out-of-sample likelihood for a new observation x_{N+1}, approaches which differ in the way the indicator z_{N+1} is estimated.

1. *Mixture predictive density.* The unknown indicator z_{N+1} can be integrated out resulting in a mixture density:

$$p\left(x_{N+1}|N_0, z, X\right) = \sum_{k=1}^{K+1} p\left(z_{N+1} = k|N_0, z, X\right) p\left(x_{N+1}|z, X, z_{N+1} = k\right) \tag{C.1}$$

   The assignment probability $p\left(z_{N+1} = k|z_N, N_0\right)$ is $\frac{N_k}{N_0+N}$ for an existing cluster and $\frac{N_0}{N_0+N}$ for a new cluster. The second term corresponds to the predictive distribution of $N+1$ point $p\left(x_{N+1}|z, X, z_{N+1} = k\right) = f\left(x_{N+1}\left|\theta_k^{-(N+1)}\right.\right)$.

2. *MAP predictive density.* We can also use a point estimate for z_{N+1} by picking the minimum negative log posterior of the indicator $p\left(z_{N+1}|x_{N+1}, N_0\right)$ or equivalently:

$$z_{N+1}^{\text{MAP}} = \underset{k\in 1,\ldots,K,K+1}{\arg\min} \left[-\ln p\left(x_{N+1}|z, X, z_{N+1} = k\right) - \ln p\left(z_{N+1} = k|N_0, z, X\right)\right] \tag{C.2}$$

   where $p\left(x_{N+1}|z, X, z_{N+1} = k\right)$ and $p\left(z_{N+1} = k|N_0, z, X\right)$ are computed as in the approach above. Once we have evaluated the MAP assignment for point $N+1$, $z_{N+1}^{\text{MAP}} = k^*$ we model x_{N+1} with predictive density $p\left(x_{N+1}|z, X, z_{N+1}^{MAP} = k^*\right) = f\left(x_{N+1}\left|\theta_{k^*}^{-(N+1)}\right.\right)$.

The first (marginalization) approach is used in (Blei & Jordan, 2006) and is more robust as it incorporates the probability mass of all cluster components while the second (modal) approach can be useful in cases where only a point prediction is needed.

Appendix D

Missing data

In MAP-DP, we can learn missing data as a natural extension of the algorithm due to its derivation from Gibbs sampling: MAP-DP can be seen as a simplification of Gibbs sampling where the sampling step is replaced with maximization. The Gibbs sampler provides us with a general, consistent and natural way of learning missing values in the data without making further assumptions, as a part of the learning algorithm. That is, we can treat the missing values from the data as latent variables and sample them iteratively from the corresponding posterior one at a time, holding the other random quantities fixed. In this framework, Gibbs sampling remains consistent as its convergence on the target distribution is still ensured. (Note that this approach is related to the ignorability assumption of (Rubin, 1976) where the missingness mechanism can be safely ignored in the modeling. Molenberghs *et al.* have shown that more complex models which model the missingness mechanism cannot be distinguished from the ignorable model on an empirical basis.)

Coming from that end, we suggest the MAP equivalent of that approach. We treat the missing values from the data set as latent variables and so update them by maximizing the corresponding posterior distribution one at a time, holding the other unknown quantities fixed. In MAP-DP, the only random quantity is the cluster indicators $z_1, \ldots, z_N$ and we learn those with the iterative MAP procedure given the observations $x_1, \ldots, x_N$. Consider some of the variables of the M-dimensional $x_1, \ldots, x_N$ are missing, then we will denote the vectors of missing values from each observations as $x_1^*, \ldots, x_N^*$ with $x_i^* = \left(x_{i,m}^*\right)_{m=1}^M$ where $x_{i,m}^*$ is empty if feature m of the observation x_i has been observed. MAP-DP for missing data proceeds as follows:

1. For each feature $m = 1, \ldots, M$, sample all of the missing values $x_{1,m}^*, \ldots, x_{N,m}^*$ from the likelihood for that variable given the prior parameters $f\left(x_i | \theta_{0,m}\right)$. Note that we assume independent priors and that the likelihood for the different variables can take different forms, as in the case study from Section 3.10.1.

2. Combine the sampled missing variables with the observed ones and proceed to update the cluster indicators $z_1, \ldots, z_N$, treating all of the variables as known. The indicators $z_1, \ldots, z_N$ are updated as above, by computing for each point i, the $K+1$ quantities $d_{i,1}, \ldots, d_{iK}, d_{i,K+1}$ and computing $z_i = \arg\min_{k \in 1,\ldots,K+1} \left[d_{i,k} - \ln N_k^{-i}\right]$.

3. Once all of the indicators $z_1, \ldots, z_N$ are updated, update the missing variables $x_1^*, \ldots, x_N^*$. For each point i, update x_i^* by taking the mode of the corresponding likelihood $x_{i,d}^* = \arg\max_{x_{\cdot,d}} f\left(x_{\cdot,d} \left| \theta_{z_i}^{-i}\right.\right)$. For the elliptical model we can take the mode of each dimension independently $x_{i,d}^* = \arg\max_{x_{\cdot,d}} f\left(x_{\cdot,d} \left| \theta_{z_{i,d}}^{-i}\right.\right)$. After all $x_1^*, \ldots, x_N^*$ are updated, go back to step 2 and update the cluster indicators $z_1, \ldots z_N$, now using the observations and the updated missing variables.

Appendix E

Estimating the model hyper parameters (θ_0, N_0)

In Bayesian models, ideally we would like to choose our hyper parameters (θ_0, N_0) from some additional information that we have for the data. This could be related to the way data is collected, the nature of the data or expert knowledge about the particular problem at hand. For instance when there is prior knowledge about the expected number of clusters, the relation $E[K^+] = N_0 \log N$ could be used to set N_0.

In cases where this is not feasible, we have considered the following alternatives:

1. *Empirical Bayes* (EB). Set the hyper parameters to their corresponding maximum marginal likelihood values. The maximum marginal likelihood expression for θ_0 will be different for the different data types and will not always be available in closed form. Usually they can be obtained from the parameter updates in Appendix A by omitting the prior terms. In MAP-DP, the maximum likelihood estimates for the hyper parameters θ_0 coincide with EB estimates as the cluster parameters θ have already been integrated out. In fact, in the simple case of conjugate exponential family models, the EB estimates and the maximum likelihood estimates for the model hyper parameters are quite similar. That is why it is common to use the maximum likelihood estimates as a simple approximation to the EB estimate. This approach is referred to as *parametric EB point estimation* (Morris, 1983). Note that using EB to learn the hyper parameter N_0 would not be efficient because there is no closed form expression for the marginal likelihood (see point 3 below, and Equation (E.1)).

2. *Multiple restarts.* Run MAP-DP with different starting values for each of the hyper parameters (θ_0, N_0), compute the NLL from Equation (3.33) including the $C(N_0, N)$ term at convergence, change one of the hyper parameters holding the rest fixed and then restart MAP-DP with the prior parameter. Set that hyper parameter to the value resulting in smallest NLL and proceed in the same way for the next hyper parameter of the model. *Bayesian optimization* (Snoek *et al.*, 2012) has also been proposed to fit model hyper parameters but requires the specification of a Gaussian Process and associated priors that may be challenging in practice. We have therefore not utilized this approach and prefer the simpler greedy search approach. However in certain cases BO may be more efficient in terms of the number of MAP-DP runs required.

3. *MAP estimate.* Place a prior on the hyper parameter of interest and numerically compute the mode of the posterior. For instance, by using a gamma prior on N_0, $p(N_0) = \text{Gamma}(a_{N_0}, b_{N_0})$, the posterior

is proportional to:

$$p\left(N_0|N,K\right) \propto \frac{\Gamma\left(N_0\right)}{\Gamma\left(N_0+N\right)} N_0^{K+a_{N_0}-1} \exp\left[-b_{N_0} N_0\right] \tag{E.1}$$

We can numerically minimize the negative log of this posterior using e.g. Newton's method. To ensure the solution is positive we can compute the gradient with respect to $\ln N_0$: as Rasmussen notes $p\left(\ln N_0|N,K^+\right)$ is log-concave and therefore has a unique maximum.

4. *Cross-validation.* By considering a finite set of values for (θ_0, N_0), choose the value corresponding to the minimum, average, out-of-sample likelihood across all cross-validation repetitions (see Appendix C). This approach is taken in (Blei & Jordan, 2006) to compare different inference methods.

We have found the second approach to be the most effective where empirical Bayes can be used to obtain the values of the hyper parameters at the first run of MAP-DP. For small datasets we recommend using the cross-validation approach as it can be less prone to overfitting.

Appendix F

Bregman divergences

The conditional probabilities for the DPMM can be expressed using the general *distortion* measure known as *Bregman divergence* (Banerjee *et al.*, 2005). The Bregman divergence between any two vectors x and θ is defined as $D_\phi(x, \theta) = \phi(x) - \phi(\theta) - \langle x - \theta, \nabla\phi(\theta)\rangle$ for the function $\phi : S \to \mathbb{R}$ being differentiable and being strictly convex on a closed convex set $S \subseteq R^D$. Bregman divergences can be efficiently used to provide a a compact parameterization of exponential family distributions with their expectation parameter. This generalizes the result that a group of points are summarized by their mean in Euclidean space to all spaces that can be described with Bregman divergence as a distortion measure.

Appendix G

DP-means λ parameter binary search

In our experiments with the DP-means algorithm, it is necessary to have an automatic way of obtaining the parameter λ for synthetic experiments where we wish to obtain a specific number of clusters K_{target}. We use a binary search approach where λ is set in a sequence of binary search steps:

1. *Initialisation*: Set λ to the mid-point of the range $\left[L_1 = 0, U_1 = M^2\right]$ where L_1, U_1 are respectively the lower and upper bounds of the range for the first iteration. M^2 is the maximal squared Euclidean distance and is set to $M^2 = \sum_{d=1}^{D} \left(\max\left(x_d\right) - \min\left(x_d\right)\right)^2$ where $\max\left(x_d\right)$, $\min\left(x_d\right)$ are respectively the upper and lower bounds of the data for dimension d. (The use of the maximal Euclidean distance originates in the DP-means algorithm step which creates a new cluster when $d_{i,k} > \lambda$ where $d_{i,k}$ is the squared Euclidean distance of data point i to the mean of cluster k.)

2. *For iteration* $i = 1, 2, \ldots$

   (a) Run the DP-means algorithm with $\lambda = \frac{1}{2}\left(U_i + L_i\right)$ which returns K_{obtained},

   (b) If $K_{\text{obtained}} > K_{\text{target}}$ then there are too many clusters so we will increase λ. We update the lower bound $L_{i+1} = \lambda$ and leave the upper bound unchanged $U_{i+1} = U_i$,

   (c) If $K_{\text{obtained}} < K_{\text{target}}$ there are too few clusters so we need to decrease λ. We update the upper bound $U_{i+1} = \lambda$ and leave the lower bound unchanged $L_{i+1} = L_i$,

   (d) Stop when $K_{\text{obtained}} = K_{\text{target}}$.

Appendix H

Gibbs sampling for DPMM (spherical Gaussian)

	Algorithm H.1: CRP-based Gibbs		**Algorithm H.2**: Fully collapsed CRP-based Gibbs
Input	$x_1, \ldots, x_N$: D-dimensional data K: number of clusters N_0: prior count σ: spherical cluster variance σ_0: prior centroid variance μ_0: prior centroid variance		$x_1, \ldots, x_N$: D-dimensional data K: number of clusters N_0: prior count σ: spherical cluster variance σ_0: prior centroid variance μ_0: prior centroid variance
Output	Posterior of indicators: $(z_1, \ldots, z_N)$ Posterior of centroids:$(\mu_1, \ldots, \mu_K)$		Posterior of indicators: $(z_1, \ldots, z_N)$
1	Initialize $z_i = 1$ for all $i \in N$	**1**	Initialize $z_i = 1$ for all $i \in N$
2	$E_{\text{new}} = \infty$	**2**	$E_{\text{new}} = \infty$
3	**repeat**	**3**	**repeat**
4	$E_{\text{old}} = E_{\text{new}}$	**4**	$E_{\text{old}} = E_{\text{new}}$
5	**for** $i \in 1, \ldots, N$	**5**	**for** $i \in 1, \ldots, N$
6	**for** $k \in 1, \ldots, K$	**6**	**for** $k \in 1, \ldots, K$
7		**7**	$\dot{\sigma}_k^{-i} = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma} N_k^{-i}\right)^{-1}$
8		**8**	$\dot{\mu}_k^{-i} = \sigma_k^{-i}\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\sum_{j:z_j=k, j\neq i} x_j\right)$
10	$d_{i,k} = \frac{1}{2\sigma}\left\|x_i - \mu_k\right\|_2^2 + \frac{D}{2}\ln\sigma - \ln N_k^{-i}$ $d_{i,K+1} = \frac{\|x-\mu_0\|_2^2}{2(\sigma+\sigma_0)} + \frac{D}{2}\ln(\sigma+\sigma_0) - \ln N_0$	**10**	$d_{i,k} = \frac{\left\|x-\dot{\mu}_k^{-i}\right\|_2^2}{2\left(\sigma+\dot{\sigma}_k^{-i}\right)} + \frac{D}{2}\ln\left(\sigma+\dot{\sigma}_k^{-i}\right) - \ln N_k^{-i}$ $d_{i,K+1} = \frac{\|x-\mu_0\|_2^2}{2(\sigma+\sigma_0)} + \frac{D}{2}\ln(\sigma+\sigma_0) - \ln N_0$
11	$d_{i,k} = \exp(-d_{i,k})$	**11**	$d_{i,k} = \exp(-d_{i,k})$
12	$z_i \sim \text{Categorical}\left(\frac{d_{i,1}}{\sum_k d_{i,k}}, \ldots, \frac{d_{i,K+1}}{\sum_k d_{i,k}}\right)$ **if** $z_i = K+1$ $\mu_{K+1} \sim \mathcal{N}\left(\mu_0 + x_i, \frac{\sigma\sigma_0}{\sigma+\sigma_0}\right)$ $K = K+1$	**12**	$z_i \sim \text{Categorical}\left(\frac{d_{i,1}}{\sum_k d_{i,k}}, \ldots, \frac{d_{i,K}}{\sum_k d_{i,k}}\right)$ **if** $z_i = K+1$ $K = K+1$
13	**for** $k \in 1, \ldots, K$	**13**	
14	$\dot{\sigma}_k = \left(\frac{1}{\sigma_0} + \frac{1}{\sigma} N_k\right)^{-1}$	**14**	
15	$\dot{\mu}_k = \dot{\sigma}_k\left(\frac{\mu_0}{\sigma_0} + \frac{1}{\sigma}\bar{x}_k\right)$	**15**	
17	$\mu_k \sim \mathcal{N}(\dot{\mu}_k, \dot{\sigma}_k)$	**17**	
19	$E_{\text{new}} = \sum_{k=1}^K \sum_{i:z_i=k} d_{i,k}$	**19**	$E_{\text{new}} = \sum_{k=1}^K \sum_{i:z_i=k} d_{i,k}$
20	**until** $(E_{\text{old}} - E_{\text{new}}) \to$ random	**20**	**until** $(E_{\text{old}} - E_{\text{new}}) \to$ random

Appendix I

Fully collapsed CRF-based Gibbs sampler

Similarly to the HDP mixtures and DP mixtures, we can integrate over both global DP measure G_0 and the local DPs $G_1, \ldots, G_K$ model directly the implicit partitioning of the sequence data through the indicator variables. The fully collapsed HDP-HMM was the original construction presented in (Beal *et al.*, 2002) as the infinite Hidden Markov model. Complete Gibbs sampler for it can be derived using the CRF-based Gibbs sampler for HDP, but has been rarely used due to the complex bookkeeping. To our knowledge quantitative comparison between the direct assignments Gibbs sampler and the CRF-based Gibbs sampler for the HDP-HMM has not yet been done. When the state parameters $\theta_1, \ldots, \theta_K$ are integrated out, the CRF-based Gibbs sampler iterates between sampling the local indicators $z_{j,t}^{local}$ and the global indicators $z_{j,c}^{global}$. In the framework of the HMMs, j is the state of point $t-1$ with t being the current point; $z_{j,t}^{local}$ points to local cluster $c \in C_j$ and $z_{j,c}^{global}$ points to the state k to which local cluster c belongs. The local clusters c here are just transitions that already exist from state j with C_j being the number of states reachable from state j (by reachable we mean to transitions that have already occurred). The global indicators can re-order the states corresponding to each c. Similarly to above, the counts $N_{j,c}$ will denote the number of existing transitions from state j to the state pointed by $z_{j,c}^{global}$(number of points in local cluster c); the counts $M_{j,k}$ denote the number of time transition from state j to state k has been drawn from the global DP, but unlike in the direct assignment sampler computing the counts of the global DP is easier as we keep explicitly the assignment variables z^{global}, $M_{j,k} = \sum_{c=1}^{C_j} \delta\left(z_{j,c}^{global}, k\right)$. For point t and for state $j \in \{1, \ldots, K\}$ that point $t-1$ can be assigned to, compute:

$$p\left(z_{j,t}^{local} = c \,\middle|\, z_{\cdot,t-1}^{local} = c^*, z_{\cdot,c^*}^{global}\right) \propto \begin{cases} N_{jc}^{-jt} p\left(x_t \,\middle|\, \theta_{z_{j,c}^{global}}^{-t}\right) & \text{for existing transition } j \to c \\ \frac{\alpha}{\sum_{k=1}^{K} M_k + M_0}\left(\sum_{k=1}^{K} M_k p\left(x_t \,|\theta_k^{-t}\right) + M_0 p\left(x_t \,|\theta_0\right)\right) & \text{for new transition from } j \end{cases} \tag{I.1}$$

where if a new transition is chosen, we need to sample its global assignment z_{j,C_j+1}^{local} using the global cluster indicator probabilities; $z_{\cdot,t-1}^{local}$ denotes simply the local cluster assignment of point $t-1$. The predictive likelihood terms $p\left(x_t \,|\theta_k^{-t}\right)$ are computed same as in the direct assignment representation and again we also decide whether to explicitly sample θ_k or integrated it, as in simpler models.

The probability in Equation (I.1) characterizes the transition mechanism given the global indicators

z^{global}. The global cluster indicators, given the local indicators on the other hand can be updated using:

$$p\left(z_{j,c}^{global}=k\left|z_{-j,c}^{global},z_{j}^{local}=c\right.\right)\propto\begin{cases}M_{k}^{-jc}\prod_{t:z_{j,t}^{local}=c}p\left(x_{t}\left|\theta_{k}^{-t}\right.\right) & \text{for existing state } k\\ M_{0}\prod_{t:z_{j,t}=c}p\left(x_{t}\left|\theta_{0}^{-t}\right.\right) & \text{for a new state}\end{cases} \tag{I.2}$$

where for the update of an global indicator $z_{j,c}^{global}$ we compute the joint likelihood of all points assigned to local cluster c (all points that have transition from j to c). Through the global indicators, we can change assignment of groups of points to a state at one time, which could lead to more efficient mixing of the MCMC. However, as mentioned above CRF-based Gibbs complicates the bookkeeping and the interpretability of the model variables. Furthermore, for applications where both computational power and memory are at premium (example of which we present in Chapter 6), the set of two indicator variables will imply more memory for storage and as indicator variables depend on the number of data points rather then just number of clusters K, their memory footprint will usually be higher then the one of component parameters depending solely on K.

Appendix J

PD-DOC experiment

Processed data The total number of patients is 527 and we have considered 215 features per patient. We have removed repeating questions from the different questioners and we have removed 4 features due to lack of their full understanding and non-comparability (Those are: MMSE-Building (name or type), Answer: 0-1(Question #13 (Form 30)); Symptoms in the past week, Answer: Non Fluctuator, Fluctuator (Question #5 (Form16)); Occupation-Current-(Question #16 (Form 2); Year of Birth (Question #21(Form 2))). Cluster 1 consists of 226 patients; Cluster 2 consists of 264 patients; Cluster 3 consists of 26 patients and Cluster 4 consists of 9 patients.

Binomial Features Significant features of Parkinson's disease from the PostCEPT/PD-DOC clinical reference data across clusters obtained using MAP-DP with appropriate distributional models for each feature. Each entry in the table is the mean score of the ordinal data in each row. Lower numbers denote condition closer to healthy. Note that the Hoehn and Yahr stage is re-mapped from { 0,1.0,1.5,2,2.5,3,4,5 } to { 0,1,2,3,4,5,6,7 } respectively. Each entry in the table is showing the mapping of a significant feature to the PD-DOC data dictionary. We have sorted the significant features in terms of their corresponding Effect Size measured by standardized mean difference.

Clinical Fluctuations-offs proportion of waking day-off*: We have possible answers 0=None, 1=1%-25% of day, 2=26%-50% of day, 3=51%-75% of day, 4=76%-100% of day.

Binary Features Significant features of Parkinson's disease from the PostCEPT/PD-DOC clinical reference data across clusters (groups) obtained using MAP-DP with appropriate distributional models for each feature. Each entry in the table is the probability of PostCEPT Parkinson's patient answering "yes" in each cluster (group). Mapping of the presented significant categorical features to the PD-DOC data dictionary has been provided. We have also include p-values from the t-test comparing Group 1 and Group 2. The odds ratio is commonly used to measure effect size for binary data. When the odds are higher then 1 , in Group 1 is more likely obtain output answer"Yes", while when the odds are smaller then 1 Group 2 is more likely to obtain output answer "No".

*GDS stands for Geriatric Depression Scale

Categorical Features Distribution of patient respond to PD state during exam. This is another significant feature which is in a separate table due to the different type of the data.

Table J.1: Binomial Data

Feature	Scale	Gr. 1	Gr. 2	Gr. 3	Gr. 4	Effect Size	p-value
Rigidity Neck	0-4	0.44	1.27	0.27	2.13	0.89	0.0000
Rigidity (left upper extremity)	0-4	0.62	1.33	0.19	2.00	0.85	0.0000
Finger Taps (Left hand)	0-4	0.65	1.41	0.50	2.33	0.83	0.0000
Hand Movements (Left hand)	0-4	0.47	1.16	0.23	2.44	0.83	0.0000
Posture	0-4	0.39	1.00	0.19	2.44	0.82	0.0000
Rapid Alternating Movements (Left hand)	0-4	0.53	1.23	0.38	2.89	0.79	0.0000
Body Bradykinesia	0-4	0.98	1.61	0.77	3.33	0.78	0.0000
Rigidity (left lower extremity)	0-4	0.38	1.06	0.04	2.67	0.77	0.0000
Dressing-ON	0-4	0.50	1.05	0.46	3.00	0.75	0.0000
Handwriting-ON	0-4	0.91	1.73	0.38	3.11	0.75	0.0000
Tremor-ON	0-4	0.93	1.52	0.81	1.33	0.72	0.0000
Cutting Food-ON	0-4	0.32	0.84	0.27	2.78	0.72	0.0000
Salivation-ON	0-4	0.34	0.87	0.54	1.67	0.71	0.0000
Leg Agility (Left leg)	0-4	0.48	1.07	0.35	2.56	0.70	0.0000
Speech-ON	0-4	0.43	1.03	0.23	1.89	0.68	0.0000
Turning in Bed-ON	0-4	0.26	0.71	0.35	2.56	0.67	0.0000
Modified Hoehn and Yahr stage	0-4	2.46	3.19	1.62	6.33	0.66	0.0000
Facial expression	0-4	1.42	1.47	0.42	2.33	0.64	0.0000
Walking-ON	0-4	0.51	0.92	0.69	2.89	0.63	0.0000
Tremor at rest (Left hand)	0-4	0.43	1.00	0.19	0.88	0.61	0.0000
Rigidity (right lower extremity)	0-4	0.46	0.97	0.04	2.56	0.59	0.0000
Speech	0-4	0.50	0.93	0.23	2.22	0.58	0.0000
Clinical Fluctuations-offs proportion of waking day-off	0-4	0.53	0.14	0.00	1.38	0.57	0.0000
Rapid Alternating Movements (Right hand)	0-4	0.58	1.05	0.15	2.33	0.57	0.0000
Hand Movements (Right hand)	0-4	0.53	0.97	0.40	2.22	0.57	0.0000
Hygiene-ON	0-4	0.26	0.59	0.12	2.44	0.56	0.0000
Action or Postural Tremor (Left hand)	0-4	0.39	0.79	0.46	1.50	0.56	0.0000
Finger Taps (Right hand)	0-4	0.75	1.20	0.36	2.33	0.53	0.0000
Tremor at rest (face, lips and chin)	0-4	0.05	0.32	0.23	1.00	0.51	0.0000
Leg Agility (Right leg)	0-4	0.46	0.84	0.19	2.44	0.50	0.0000
Rigidity (right upper extremity)	0-4	0.90	1.30	0.38	2.11	0.49	0.0000
Gait	0-4	0.30	0.62	0.23	2.89	0.49	0.0000
Freezing when walking-ON	0-4	0.06	0.29	0.12	1.78	0.41	0.0000
Swallowing-ON	0-4	0.08	0.29	0.19	0.78	0.41	0.0000
Postural Stability	0-4	0.07	0.31	0.19	3.22	0.39	0.0000
Action or Postural Tremor (Right hand)	0-4	0.41	0.70	0.62	1.38	0.38	0.0000
Tremor at rest (Left foot)	0-4	0.11	0.29	0.00	0.63	0.36	0.0001
Arising from Chair	0-4	0.11	0.33	0.15	3.22	0.35	0.0000
Intellectual Impairment-ON	0-4	0.26	0.49	0.38	2.22	0.35	0.0000
Tremor at rest (Right foot)	0-4	0.70	1.00	0.35	1.11	0.30	0.0009
Motivation/Initiation-ON	0-4	0.27	0.48	0.58	2.22	0.29	0.0006
Tremor at rest (Right foot)	0-4	0.11	0.26	0.00	0.50	0.29	0.0012
MMSE Calculated Total	0-30	29.3	28.9	28.7	23.2	0.23	0.0033
Falling-ON	0-4	0.01	0.09	0.12	1.11	0.22	0.0014

Table J.2: Binary Data

Feature	Gr. 1	Gr. 2	Gr. 3	Gr. 4	Odds ratio	p-value
Clinical Fluctuations-offs come one suddenly	0.04	0.004	0.00	0.00	10.38	0.0049
Clinical Fluctuations-offs predictable	0.43	0.08	0.00	0.33	8.68	0.0000
Resting tremor (absent)	0.14	0.06	0.42	0.11	2.55	0.0068
Gender	0.40	0.29	0.42	0.56	1.63	0.0067
GDS* Have more problems with memory then most	0.16	0.26	0.19	0.89	0.54	0.0056
GDS* Dropped many activities	0.10	0.19	0.19	0.89	0.47	0.0058
Resting tremor (present and typical)	0.81	0.91	0.42	0.78	0.42	0.0013

Table J.3: Categorical Data

PD state during exam	Gr. 1	Gr. 2	Gr. 3	Gr. 4	p-value
Fluctuator-ON during the exam	0.44	0.05	0.00	0.33	0.0000
Fluctuator- Fluctuated during the exam	0.01	0.004	0.00	0.00	0.0000
Fluctuator-OFF during exam	0.01	0.008	0.00	0.00	0.0000
Non-fluctuator	0.54	0.94	1.00	1.67	0.0000

Poisson Features Age at Baseline is also a significant feature. It has been separated from the other features as it is Poisson data type and sorting in terms of Effect size might be misleading.

Table J.4:

Feature	Gr. 1	Gr. 2	Gr. 3	Gr. 4	Effect Size	p-value
Age at BaseLine	60.53	64.38	63.39	73.44	0.39	0.0000