Structured Bayesian Learning Through Mixture Models

by

Francesca Petralia

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
David B. Dunson, Supervisor

_____
David L. Banks

_____
Surya T. Tokdar

_____
Joseph Lucas

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2013

Abstract

Structured Bayesian Learning Through Mixture Models

by

Francesca Petralia

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
David B. Dunson, Supervisor

_____
David L. Banks

_____
Surya T. Tokdar

_____
Joseph Lucas

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2013

Abstract

In this thesis, we develop some Bayesian mixture density estimation for univariate and multivariate data. We start proposing a repulsive process favoring mixture components further apart. While conducting inferences on the cluster-specific parameters, current frequentist and Bayesian methods often encounter problems when clusters are placed too close together to be scientifically meaningful. Current Bayesian practice generates component-specific parameters independently from a common prior, which tends to favor similar components and often leads to substantial probability assigned to redundant components that are not needed to fit the data. As an alternative, we propose to generate components from a repulsive process, which leads to fewer, better separated and more interpretable clusters.

In the second part of the thesis, we face the problem of modeling the conditional distribution of a response variable given a high dimensional vector of predictors potentially concentrated near a lower dimensional subspace or manifold. In many settings it is important to allow not only the mean but also the variance and shape of the response density to change flexibly with features, which are massive-dimensional. We propose a multiresolution model that scales efficiently to massive numbers of features, and can be implemented efficiently with slice sampling.

In the third part of the thesis, we deal with the problem of characterizing the conditional density of a multivariate vector of response given a potentially high dimensional vector of predictors. The proposed model flexibly characterizes the density

of the response variable by hierarchically coupling a collection of factor models, each one defined on a different scale of resolution. As it is illustrated in Chapter 4, our proposed method achieves good predictive performance compared to competitive models while efficiently scaling to high dimensional predictors.

to My Parents

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathcal{X}$	Space of predictors
$\mathcal{Y}$	Space of response variable
x	Vector of predictors
y	Response variable
$\Re$	Real numbers
$Dirichlet$	Dirichlet density.
Un	Uniform density.
$\mathcal{IG}$	Inverse Gamma density
$\mathcal{N}(\mu, \sigma)$	univariate Normal density with mean μ and variance σ
$\mathcal{N}_p(\mu, \Sigma)$	p-variate Normal density with mean μ and covariance matrix Σ
$Mult$	Multinomial density

Abbreviations

MSE	Mean Squared Error.
N-R	Non-Repulsive.
R	Repulsive
MSB	Multiresolution Stick Breaking
BFT	Bayesian Factor Tree
MFA	Mixture of factor analyzers

dMFA	Covariates dependent Mixture of factor analyzers
K-L	Kullback-Leibler
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
i.i.d.	Independent and identically distributed
CART	Classification and regression trees
RF	Random forest
CPU	Central processing unit

Acknowledgements

The completion of this work would have not been possible without the guidance and advice of my advisor David Dunson. He helped me to grow as a researcher and to keep things in perspective. At the same time, he motivated me a lot with his enthusiasm towards statistics and his curiosity about other disciplines and new problems. I am very grateful to my thesis committee members: Surya Tokdar, David Banks and Joseph Lucas. A very special thanks to Mike West and David Banks. They helped me start my experience at Duke and encouraged me during the difficult moments. I really enjoyed collaborating with scientists from other fields. In particular I want to mention Joshua Vogelstein and Vinayak Rao.

1

Introduction

1.1 Motivation

For decades mixture models have been extensively used for classification, discrimination and density estimation. In analyses of finite mixture models, a common concern is over-fitting in which redundant mixture components having similar locations and scales are introduced. Over-fitting can have an adverse impact on density estimation, since this leads to an unnecessarily complex model. Another common goal of finite mixture modeling is clustering (Fraley and Raftery, 2002), and having components with similar locations, leads to overlapping kernels and lack of interpretability. Introducing kernels with similar locations but different scales may be necessary to fit heavy-tailed and skewed densities, and hence low separation in clustering and over-fitting are distinct problems.

Recently, Rousseau and Mengersen (2011) studied the asymptotic behavior of the posterior distribution in over-fitted Bayesian mixture models having more components than needed. They showed that a carefully chosen prior will lead to asymptotic emptying of the redundant components. However, several challenging practical issues arise. For small to moderate sample sizes, the weight assigned to redundant

components is often substantial. This can be attributed to identifiability problems that arise from a difficulty in distinguishing between models that partition each of a small number of well separated components into a number of essentially identical components. This issue leads to substantial uncertainty in clustering and estimation of the number of components, and is not specific to over-fitted mixture models; similar behavior occurs in placing a prior on the number of components or using a nonparametric Bayes approach such as the Dirichlet process.

The problem of separating components has been studied for Gaussian mixture models (Dasgupta, 1999; Dasgupta and Schulman, 2007). Two Gaussians can be separated by placing an arbitrarily chosen lower bound on the distance between their means. Separated Gaussians have been mainly utilized to speed up convergence of the Expectation-Maximization (EM) algorithm. In choosing a minimal separation level, it is not clear how to obtain a good compromise between values that are too low to solve the problem and ones that are so large that one obtains a poor fit. Alternatively, we propose a repulsive prior discouraging closeness among component-specific parameters without placing an hard constraint. This repulsive process leads to better separated and more interpretable clusters while accurately estimating the true density of the data.

Mixture models are also utilized to describe the conditional distribution of a response variables given a set of predictors. In this framework an important issue is the scalability of mixture models to massive numbers of predictors. Massive datasets present statistical and computational challenges for machine learning because many previously developed approaches do not scale-up sufficiently. Specifically, challenges arise because of the ultrahigh-dimensionality, and relatively low sample size (the "large p, small n" problem, Bernardo et al. (2003)). Parsimonious models for such big data assume that the density in the ambient dimension concentrates around a lower-dimensional and possibly nonlinear subspace. Indeed, a plethora of methodologies

are emerging to estimate such lower-dimensional manifolds from high-dimensional data (Rahman et al., 2005; Allard et al., 2012).

There is a rich machine learning and statistical literature on conditional density estimation of a response $y \in \mathcal{Y}$ given a set of features (predictors) $x = (x_1, x_2, \ldots, x_p) \in \mathcal{X}$. Common approaches include hierarchical mixtures of experts (Jacobs et al., 1991; Jiang and Tanner, 1999), kernel methods (Fan et al., 1996; Fan and Yim, 2004; Holmes et al., 2010; Fu et al., 2011), Bayesian finite mixture models (Nott et al., 2012; Tran et al., 2012; Norets and Pelenis, 2012) and Bayesian nonparametrics (Griffin and Steel, 2006; Dunson et al., 2007; Chung and Dunson, 2009; Tokdar et al., 2010). In all these works, there has been limited consideration of scaling to large p settings, with the variational Bayes approach of Tran et al. (2012) being a notable exception. For dimensionality reduction, they follow a greedy variable selection algorithm. Their approach does not scale to the sized applications we are interested in. For example, in a problem with $p = 1,000$ and $n = 500$, they reported a CPU time of 51.7 minutes for a single analysis. We are interested in problems many orders of magnitude or more larger than this, and require a faster computing time while also accurately estimating the conditional density of a response variable. To our knowledge, there are no nonparametric density regression competitors to our approach, which maintain a characterization of uncertainty in estimating the conditional densities; rather, all sufficiently scalable algorithms provide point predictions and/or rely on restrictive assumptions such as linearity.

A widely used method to estimate a covariates-dependent density is to partition observations into a nested sequence of subsets based on feature similarity, with simple models fit within each subset. This is the basis for CART (Breiman et al., 1984), modifications such as random forests (Breiman, 2001), boosting (Shapire et al., 1998) and bagging (Breiman, 1996). Though these algorithms can substantially improve mean square error performance, computation can be expensive and performance de-

grades as the dimensionality of the predictor space increases. In fact, a significant downside of divide-and-conquer algorithms is their poor scalability to high dimensional predictors. As the number of features increases, the problem of finding the best splitting attribute becomes intractable so that tree based models cannot be efficiently applied. As the number of features increases, also mixture of experts models become computationally demanding, since both mixture weights and dictionary densities are feature-dependent. In an attempt to make mixtures of experts more efficient, sparse extensions relying on different variable selection algorithms have been proposed (Mossavat and Amft, 2011). However, performing variable selection in high dimensions is effectively intractable: algorithms need to efficiently search for the best subsets of predictors to include in weight and mean functions within a mixture model, an NP-hard problem. In chapter 3 we propose an algorithm based on a novel stick breaking process which can scale substantially better than competitors to high dimensional predictors while efficiently estimating the conditional density of a response variable.

In this thesis we also focus on the challenging problem of learning a multivariate density of a vector $y \in \mathcal{Y} \subseteq \Re^p$ indexed by features $x \in \mathcal{X} \subseteq \Re^q$. This is an important problem in many domains. For example, one may want to learn the joint density of brain activity across sensors from MEG, EEG or fMRI data as a function of patient tasks and characteristics. In modeling such data, it is most common to assume either independence across sensors or that the data are multivariate Gaussian, with the emphasis then on estimating a covariates dependent mean vector $\mu(x)$ and covariance matrix $\Sigma(x)$ (Fyshe et al., 2012). Though flexible approaches have been introduced to model the $p \times p$ feature-dependent covariance matrix (Pourahmadi, 1999; Chiu et al., 1996; Hoff and Niu, 2012; Gelfand et al., 2004; Williams, 1996), the predictive performance of such models depends strictly on the validity of the normality assumption.

There has been relatively limited attention on multivariate conditional density estimation. Notable exceptions include Krauthausen and Hanebeck (2010) and Davis and Hwang (1998). Krauthausen and Hanebeck (2010) models y through a mixture of spherical Gaussians with feature-dependent weights, while Davis and Hwang (1998) estimates $f(y|x)$ by placing kernels with varying bandwidths at each of the training data points. These algorithms have been tested only on low dimensional datasets, and may become computationally intractable in bigger problems. Tree based models, typically applied to settings involving a univariate response but can be easily implemented in multivariate settings (Death, 2002; Larsen and Speckman, 2004; Hothorn et al., 2006; Lutz and Buhlmann, 2006). Although performance is often excellent in small to moderate dimensions, scaling to large numbers of features is a general problem for usual tree-based models.

1.2 Literature Review

1.2.1 Mixture Models

Finite mixture models characterize the density of $y \in \mathcal{Y} \subseteq \Re^m$ as

$$f(y|p,\gamma) = \sum_{h=1}^{k} p_h \phi(y;\gamma_h), \tag{1.1}$$

where $p = (p_1, \ldots, p_k)^T$ is a vector of probabilities summing to one, and $\phi(\cdot;\gamma)$ is a kernel depending on parameters $\gamma \in \Gamma$, which may consist of location and scale parameters (McLachlan and Peel, 2000). There is a very rich literature on inference for finite mixture models from both a frequentist (Figueiredo and Jain, 2002; Muthen and Shedden, 1999) and Bayesian (Richardson and Green, 1997) perspective. In practice, most of the frequentist literature focuses on maximum likelihood estimation, with the Akaike information criterion (AIC) and other criteria used to estimate the number of mixture components (Raftery and Fraley, 1998). Bayesian approaches

instead rely on placing a prior on the number of components and the component-specific parameters, and hence may have some advantages in terms of accounting for uncertainty in estimating the number of components while also regularizing the component-specific parameters (Escobar and West, 1995; Richardson and Green, 1997). However, due to ease in computation, it has become popular to use over-fitted mixture models in which a conservative upper bound on the number of components is chosen.

Considering the finite mixture model in expression (1.1), a Bayesian specification is completed by choosing priors for the number of components k, the probability weights p, and the component-specific parameters $\gamma = (\gamma_1, \ldots, \gamma_k)^T$. Typically, k is assigned a Poisson or multinomial prior, p a $Dirichlet(\alpha)$ prior with $\alpha = (\alpha_1, \ldots, \alpha_k)^T$, and $\gamma_h \sim P_0$ independently, with P_0 often chosen to be conjugate to the kernel ϕ. As an example, when ϕ is the normal kernel and γ is a vector containing mean and standard deviation, i.e. $\gamma = (\mu, \sigma)^T$, a normal inverse-Gamma prior is assigned to γ.

Posterior computation can proceed via a reversible jump Markov chain Monte Carlo (Richardson and Green, 1997) algorithm involving moves for adding or deleting mixture components. Unfortunately, in making a $k \rightarrow k+1$ change in model dimension, efficient moves critically depend on the choice of proposal density. Stephens (2000a) proposed an alternate Markov chain Monte Carlo method, which treats the parameters as a marked point process, but does not have clear computational advantages relative to reversible jump. For these reasons It has become popular to use over-fitted mixture models in which k is chosen as a conservative upper bound on the number of components. From a practical perspective, the success of over-fitted mixture models has been largely due to ease in computation.

As motivated in Ishwaran and Zarepour (2002), simply letting $\alpha_h = c/k$ for $h \in \{1, \ldots, k\}$ and a constant $c > 0$ leads to an approximation to a Dirichlet process

mixture model for the density of y, which is obtained in the limit as k approaches infinity. An alternative finite approximation to a Dirichlet process mixture is obtained by truncating the stick-breaking representation of Sethuraman (1994a), leading to a similarly simple Gibbs sampling algorithm (Ishwaran and James, 2001). These approaches are now used routinely in practice.

When working with mixture models, an important issue is the identifiability of mixture parameters (γ, p). In general, parameters are identifiable if distinct parameter values lead to different densities. Let $f(y|\theta)$ be a mixture density defined as in 1.1 with θ being the vector of mixture parameters, i.e. $\theta = (\gamma^T, p^T)^T$. It can be easily shown that two different vectors θ and θ' can lead to the same mixture density, i.e. $f(y|\theta) = f(y|\theta')$. The lack of identifiability is mainly caused by the invariance of the likelihood to relabeling of the components and over-fitting. The first identifiability issue does not create problems when the model is estimated through maximum likelihood; however it causes major problems when dealing with Bayesian methods based on Markov chains Monte Carlo. In the Bayesian literature the lack of identifiability due to relabeling of the components is a challenging problem better known as label switching problem (Stephens, 2000b; Lavine and West, 1992; Jasra et al., 2005). An effective technique used to overcome this identifiability problem is relabeling the clusters at each MCMC iteration using a post-processing algorithm. Examples of such approach include Yao and Lindsay (2009), Stephens (2000b) and Cron and West (2011). A more serious identifiability issue is due to the introduction of equal components. As an example, consider two mixture models $f(y|p^{(k)}, \gamma^{(k)})$ and $f(y|p^{(k+1)}, \gamma^{(k+1)})$ involving k and $k+1$ components respectively. It can be easily shown that parameters $(p^{(k)}, \gamma^{(k)})$ and $(p^{(k+1)}, \gamma^{(k+1)})$ satisfying the following constraints

$$p_h^{(k+1)} = p_h^{(k)}, \quad \gamma_h^{(k+1)} = \gamma_h^{(k)}, \quad \forall h < k$$

$$p_k^{(k)} = p_{k+1}^{(k+1)} + p_k^{(k+1)} \quad , \quad \gamma_{k+1}^{(k+1)} = \gamma_k^{(k+1)}$$

lead to the same mixture density, i.e. $f(y|p^{(k)}, \gamma^{(k)}) = f(y|p^{(k+1)}, \gamma^{(k+1)})$. This identifiability issue may result in over-fitted mixtures where identical and consequently unnecessary components are introduced. To our knowledge no methods have been proposed to solve this identifiability issue.

1.2.2 Divide and Conquer Algorithms and Tree Based Models

Divide and conquer algorithms fit surfaces to data by explicitly dividing the input space into a nested sequence of regions, and by fitting simple surfaces within these regions. A well known example of such algorithms is CART (classification and regression trees) (Breiman et al., 1984). Starting from a set including all observations (root), tree based methods recursively splits each subset into subsets containing more homogenous observations. Generally observations are allocated to different subsets through greedy algorithms and the number of subsets is determined by pruning the tree according to a model choice criterion such as AIC and BIC. Recently, tree based methods relying on full Bayesian specifications have been introduced. Bayesian tree models estimate the tree by placing a prior on the space of all trees and implementing stochastic search algorithms to explore the entire space (Chipman et al., 1993; Wu et al., 2007; Mallick, 1998).

Though CART models are appealing in providing a simple, flexible and interpretable mechanism of dimensionality reduction, it is well known that single tree estimates commonly have high variance and poor performance. There is a rich machine learning literature proposing improvements based on bagging (Breiman, 1996), boosting (Shapire et al., 1998) and random forests (Breiman, 2001). All these methods overcome the limit associated to single tree models by combining results generated from multiple trees. The multiple trees setup can certainly leads to better mean square errors by reducing the variability associated to the estimates. However,

these approaches may become computationally intensive when dealing with massive number of features.

Another divide-and-conquer algorithm particularly useful to reduce the variance associated to single tree estimates is mixture of experts (Jacobs et al., 1991). As opposed to other divide-and-conquer algorithms, mixture of experts relies on soft partitioning algorithms that allows observations to lie simultaneously in different subsets. A mixture of experts model is a mixture model in which the model parameters, including mixture weights, are functions of covariates. In practice, observations are assigned to different experts by a gating network through a probabilistic model. Then, within each expert, observations are considered identically distributed. A variety of mixture of experts models have been proposed in the last twenty years. Some of them deal with infinitely many experts (Rasmussen and Ghahramani, 2002; Meeds and Osindero, 2006), others propose a hierarchical structure where the density within each expert is a mixture model (Jordan and Jacobs, 1994; Bishop and Svensen, 2003).

1.2.3 Factor Model and Mixture of Factor Analyzers

Factor analysis has been one of the most flexible tools utilized to model the dependence structure of a p-dimensional vector of random variables through a sparse decomposition of a $p \times p$ covariance matrix, $\Sigma = \Theta\Theta^T + \Sigma_0$ with $\Sigma_0 = diag(\sigma_1, \ldots, \sigma_p)$. This covariance decomposition is obtained by considering the following model for $y_i \in \Re^p$

$$y_i = \mu_0 + \Theta\eta_i + \epsilon_i \qquad \eta_i \sim \mathcal{N}_k(0, I) \qquad \epsilon_i \sim \mathcal{N}_p(0, \Sigma_0) \tag{1.2}$$

with Θ being a $p \times k$ loading matrix and $k << p$. Model 1.2 implies that the elements of y_i are conditionally independent given the latent factors and the marginal dependence among them is induced by the shared dependence on the latent factors. The covariance matrix Σ can be derived by marginalizing out η_i. Tipping and Bishop

(2012) with their probabilistic principal component analysis showed that, under an isotropic error model, i.e. $\Sigma_0 = \sigma I$, the maximum likelihood estimate of the k columns of the loading matrix converges to the first k principal components of the data as σ approaches zero. Therefore, considering an isotropic error, it is possible to combine the advantages of a probabilistic model with those of principal component analysis.

Often there is interest in estimating the latent factors, interpreted as underlying processes characterizing the data. However, these factors are not identifiable without imposing further constraints on the loading matrix (Bernardo et al., 2003; Lopes and West, 2004). In fact, for any $k \times k$ orthogonal matrix Γ, Θ and $\Theta' = \Theta\Gamma$, lead to the same covariance decomposition. To solve this identifiability issue and uniquely estimate the latent factors one could constrain the loading matrix to be lower triangular (Geweke and Zhou, 1996; Aguilar and West, 2000) or orthogonal (Seber, 2004). Though inference on latent factors remains an interesting and open problem, in many applications the main focus is the estimation of the covariance matrix Σ. Latent factor models provide a low rank approximation of a large scale covariance matrix and it is related to a set of articles, including Zou et al. (2006), Shen and Huang (2008), Witten et al. (2009) and Johnstone and Lu (2009), which mainly focused on sparse principal component analysis. In the analysis of factor models another crucial point is the determination of the number of factors. The number of latent factors can be determined through variable selection criteria (Onatski, 2005; Minka, 2001), reversible jump algorithms (Lopes et al., 2011; Hastie and Green, 2012) and adaptive Markov chains (Bhattacharya and Dunson, 2011).

Though factor model offers a flexible tool to describe the dependence structure of a set of variables its applicability is limited by linearity. This limitation can be overcome by combining local models in the form of finite mixture (Tipping and Bishop, 1997). Mixtures of factor analyzers (MFA) model a p-dimensional vector of

observations as follows

$$y_i \sim \sum_{h=1}^{N} p_h \mathcal{N}_p \left(\mu_h, \Theta_h \Theta_h' + \Sigma_{h0}\right) \tag{1.3}$$

with $\mu_h \in \Re^p$, $\Theta_h \in \Re^{p \times k}$ being the loading matrix, $(p_1, \ldots, p_N)^T$ being positive weights summing up to one, $\Sigma_{h0} = diag(\sigma_{h1}, \ldots, \sigma_{hp})$. According to model 4.1, observations are assumed to belong to the hth cluster with probability p_h and, within each cluster, observations are modeled through a linear factor model (equation 1.2) with parameters $(\mu_h, \Lambda_h, \Sigma_{h0})$. Mixture of factor analyzers offers the potential to adequately model the density of high-dimensional observations while also allowing for both clustering and local dimensionality reduction. In order to estimate the parameters of the MFA many approaches have been introduced. Some of them (Ghahramani and Hinton, 1997; Zhou and Liu, 2008) estimate the model using the Expectation Maximization algorithm (Dempster et al., 1977), others rely on variational inference (Ghahramani and Beal, 2000), others on full bayesian inference (Utsugi and Kumagai, 2011).

1.3 Dissertation Outline

In this thesis we deal with different problems in mixture modeling such us identifiability, over-fitting and scalability to massive number of features.

The second chapter offers a possible solution to the identifiability and over-fitting problem characterizing finite mixture models. In contrast to the majority of the Bayesian literature on discrete mixture models, instead of drawing the component-specific parameters $\{\gamma_h\}$ in 1.1 independently from a common prior, we propose a joint prior for $\gamma = (\gamma_1, \ldots, \gamma_k)^T$ that is chosen to assign low density to γ_h's located close together. We consider two types of repulsive priors, (i) priors guarding against over-fitting by penalizing redundant kernels having close to identical locations and

scales and case (ii) priors discouraging closeness in only the locations to favor well separated clusters.

The third chapter focuses on learning the conditional density of a response variable given an high dimensional vector of predictors. We present a multiresolution approach which learns a multiscale dictionary of densities, constructed as Gaussian within each set of a multiscale partition tree for the features. This tree is efficiently learned in a first stage using a fast and scalable graph partitioning algorithm (Karypis and Kumar, 1999). Then, the conditional density $f(y|x)$ for each $x \in \mathcal{X}$ is expressed as a convex combination of coarse to fine scale dictionary densities. This is accomplished in a Bayesian manner using a novel multiresolution stick-breaking process, which allows the data to inform about the optimal bias-variance tradeoff. The proposed model allows borrowing information across different resolution levels and reaches a good compromise in terms of the bias-variance tradeoff. We show that the algorithm scales efficiently to massive numbers of features.

Finally, the fourth chapter focuses on learning the conditional density of a multivariate vector of response given an high dimensional vector of predictors. In many applications, there is interest in assessing how the density of a multivariate response changes as function of features, with both the response and the predictor being highly dimensional. To address this challenging problem, we propose a multiscale predictor-dependent mixture of factor analyzers in which specific-component parameters depend on the path of the predictor vector through a multiscale partition tree. By borrowing information across resolution levels, we allow local adaptivity in which a single factor model may suffice in terms of the bias-variance tradeoff in certain regions of the predictor space, while in other regions additional layers are required.

2

Repulsive Mixtures

Mixture models have been extensively utilized for density estimation, clustering and as a component in flexible hierarchical models. In using mixture models for clustering, identifiability problems arise if mixture components are not sufficiently well separated and the data for the different sub-populations contain substantial overlap. Insufficiently separated components also create problems in using mixture models for density estimation and robust modeling, as redundant components that are located close together can be introduced leading to an unnecessarily complex model as well as to various computational problems. Current practice in Bayesian mixture modeling generates the component-specific parameters from a common prior, which tends to favor components that are close together. As an alternative, in this chapter, we propose to generate mixture components from a repulsive process that favors placing components further apart.

2.1 Bayesian Repulsive Mixture Models

2.1.1 Repulsive Densities

We seek a prior on the component parameters in (1.1) that automatically favors spread out components near the support of the data. Instead of generating the atoms γ_h independently from P_0, one could generate them from a repulsive process that automatically pushes the atoms apart. This idea is conceptually related to the literature on repulsive point processes (Huber and Wolpert, 2009). In the spatial statistics literature, a variety of repulsive processes have been proposed. One such model assumes that points are clustered spatially, with the vector of cluster centers γ having a Strauss density (Lawson and Clark, 2002), that is $p(k, \gamma) \propto \beta^k \rho^{r(\gamma)}$ where k is the number of clusters, $\beta > 0$, $0 < \rho \leqslant 1$ and $r(\gamma)$ is the number of pairwise centers that lie within a pre-specified distance r of each other. A possibly unappealing feature is that repulsion is not directly dependent on the pairwise distances between the clusters. We propose an alternative class of priors, which smoothly push apart components based on their pairwise distances.

Def 1. *A density $h(\gamma)$ is repulsive if for any $\delta > 0$ there is a corresponding $\epsilon > 0$ such that $h(\gamma) < \delta$ for all $\gamma \in \Gamma \backslash G_\epsilon$, where $G_\epsilon = \{\gamma : d(\gamma_s, \gamma_j) > \epsilon; s = 1, \ldots, k; j < s\}$ and d is a distance.*

We consider two special cases (i) $d(\gamma_s, \gamma_j)$ is the distance between the sth and jth kernel, (ii) $d(\gamma_s, \gamma_j)$ is the distance between sub-vectors of γ_s and γ_j corresponding to only locations. Priors following definition 1(i) limit over-fitting in density estimation, while priors following definition 1(ii) favor well-separated clusters.

As a convenient class of repulsive priors which smoothly push components apart, we propose

$$\pi(\gamma) = c_1 \left(\prod_{j=1}^{k} g_0(\gamma_j) \right) h(\gamma), \tag{2.1}$$

with c_1 being a normalizing constant that can be intractable to calculate. The dependence of c_1 on k leads to complications in estimating k that motivate the use of an over-specified mixture that treats k as an upper bound on the number of components. The proposed prior is closely related to a class of point processes from the statistical physics and spatial statistics literature called Gibbs processes (Daley and Vere-Jones, 2008). We assume $g_0 : \Gamma \to \Re_+$ and $h : \Gamma^k \to [0, \infty)$ are continuous with respect to Lesbesgue measure, and h is bounded above by a positive constant c_2 and is repulsive according to definition 1 with d differing across cases. It follows that density π defined in (2.1) is also repulsive. For location-scale kernels, let $\gamma_j = (\mu_j, \Sigma_j)$ and $g_0(\mu_j, \Sigma_j) = \xi(\mu_j)\psi(\Sigma_j)$ with μ_j and Σ_j being respectively the location and the scale parameters. A special hardcore repulsion is produced if the repulsion function is zero when at least one pairwise distance is smaller than a pre-specified threshold. Such a density implies choosing a minimal separation level between the atoms.

We avoid hard separation thresholds by considering repulsive priors that smoothly push components apart. In particular, we propose two repulsion functions defined as

$$h(\gamma) = \prod_{\{(s,j)\in A\}} g\{d(\gamma_s, \gamma_j)\} \qquad (2.2) \qquad h(\gamma) = \min_{\{(s,j)\in A\}} g\{d(\gamma_s, \gamma_j)\} \qquad (2.3)$$

with $A = \{(s, j) : s = 1, \ldots, k; j < s\}$ and $g : \Re_+ \to [0, M]$ a strictly monotone differentiable function with $g(0) = 0$, $g(x) > 0$ for all $x > 0$ and $M < \infty$. It is straightforward to show that h in (2.2) and (2.3) is integrable and satisfies definition 1. The two alternative repulsion functions differ in their dependence on the relative distances between components, with all the pairwise distances playing a role in (2.2), while (2.3) only depends on the minimal separation. A flexible choice of g corresponds to

$$g\{d(\gamma_s, \gamma_j)\} = \exp\big[- \tau\{d(\gamma_s, \gamma_j)\}^{-\nu}\big], \qquad (2.4)$$

where $\tau > 0$ is a scale parameter and ν is a positive integer controlling the rate at

which g approaches zero as $d(\gamma_s, \gamma_j)$ decreases. Figure 2.1 shows contour plots of the prior $\pi(\gamma_1, \gamma_2)$ defined as (2.1) and satisfying definition 1(ii) with $\gamma_1, \gamma_2 \in \mathbb{R}$, d the Euclidean distance, g_0 the standard normal density, the repulsive function defined as (2.2) or (2.3) and g defined as (2.4) for different values of (τ, ν). As τ and ν increase, the prior increasingly favors well separated components.

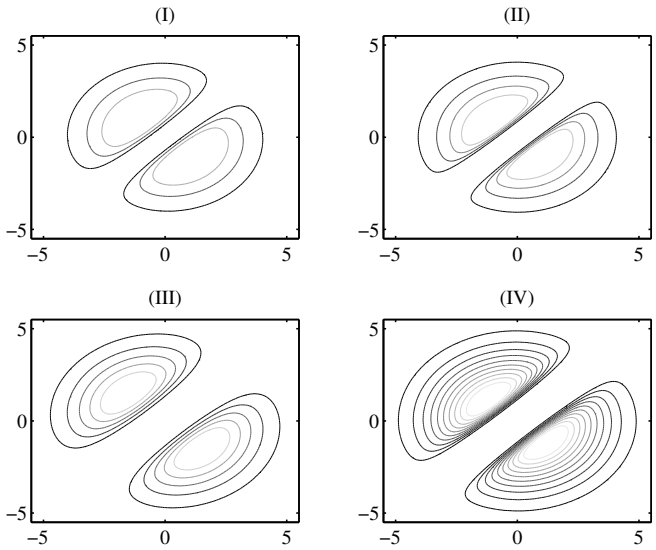

FIGURE 2.1: Contour plots of the repulsive prior $\pi(\gamma_1, \gamma_2)$ satisfying definition 1(ii) under (2.1) either (2.2) or (2.3) and (2.4) with hyperparameters (τ, ν) equal to $(I)(1,2), (II)(1,4), (III)(5,2)$ and $(IV)(5,4)$

2.1.2 Theoretical Properties

Theoretical properties of the proposed prior are considered under definition 1(ii), though all results can be modified to accommodate definition 1(i). For some results, the kernel will be assumed to depend only on location parameters, while for others on both location and scale parameters. Let Π be the prior induced on $\bigcup_{j=1}^{\infty} \mathcal{F}_k$, where $\mathcal{F}_k$ is the space of all distributions defined as (1.1). Let $\|\cdot\|_1$ denote the L_1 norm and $KL(f_0, f) = \int f_0 \log(f_0/f)$ refer to the Kullback-Leibler (K-L) divergence between f_0 and f. Density f_0 belongs to the K-L support of the prior Π if $\Pi\{f :$

$KL(f_0, f) < \epsilon\} > 0$ for all $\epsilon > 0$. Let the true density $f_0 : \Re^m \to \Re_+$ be defined as $f_0 = \sum_{h=1}^{k_0} p_{0h}\phi(\gamma_{0h})$ with $\gamma_{0h} \in \Gamma$ and γ_{0j}s such that there exists an $\epsilon_1 > 0$ such that $\min_{\{(s,j):s<j\}} d(\gamma_{0s}, \gamma_{0j}) \geqslant \epsilon_1$, d being the Euclidean distance of sub-vectors of γ_{0j} and γ_{0s} corresponding to only locations. Let $f = \sum_{h=1}^{k} p_h\phi(\gamma_h)$ with $\gamma_h \in \Gamma$. Let $\gamma \sim \pi$ and π satisfy definition 1(ii). Let $p \sim \lambda$ with $\lambda = Dirichlet(\alpha)$ and $k \sim \vartheta$ with $\vartheta(k = k_0) > 0$. Let $\theta = (p, \gamma)$. These assumptions on f_0 and f will be referred to as condition B0. The next lemma provides sufficient conditions under which the true density is in the K-L support of the prior for location kernels.

Lemma 2. *Assume condition B0 is satisfied with $m = 1$. Let D_0 be a compact set containing location parameters $(\gamma_{01}, \ldots, \gamma_{0k_0})$. Let ϕ and π satisfy the following conditions:*

A1. for any $y \in \mathcal{Y}$, the map $\gamma \to \phi(y; \gamma)$ is uniformly continuous

A2. for any $y \in \mathcal{Y}$, $\phi(y; \gamma)$ is bounded above by a constant

A3. $\int f_0 \left|\log\left\{\sup_{\gamma\in D_0} \phi(\gamma)\right\} - \log\left\{\inf_{\gamma\in D_0} \phi(\gamma)\right\}\right| < \infty$

A4. π is continuous with respect to Lebesgue measure and for any vector $x \in \Gamma^k$ with $\min_{\{(s,j):s<j\}} d(x_s, x_j) \geqslant \upsilon$ for $\upsilon > 0$ there is a $\delta > 0$ such that $\pi(\gamma) > 0$ for all γ satisfying $||\gamma - x||_1 < \delta$

Then f_0 is in the K-L support of the prior Π.

Lemma 3. *The repulsive density in (2.1) with h defined as either (2.2) or (2.3) satisfies condition A4 in lemma 2.*

The next lemma formalizes the posterior rate of concentration for univariate location mixtures of Gaussians.

Lemma 4. *Let condition B0 be satisfied, let $m = 1$ and ϕ be the normal kernel depending on a location parameter μ and a scale parameter σ. Assume that condition $(i), (ii)$ and (iii) of theorem 3.1 in Scricciolo (2011) and assumption A4 in lemma 2 are satisfied. Furthermore, assume that*

C1) the joint density π leads to exchangeable random variables and for all k the marginal density of μ_1 satisfies $\pi_m(|\mu_1| \geqslant t) \lesssim \exp(-q_1 t^2)$ for a given $q_1 > 0$

C2) there are constants $u_1, u_2, u_3 > 0$, possibly depending on f_0, such that for any $\epsilon \leqslant u_3$

$$\pi(||\mu - \mu_0||_1 \leqslant \epsilon) \geqslant u_1 \exp(-u_2 k_0 \log(1/\epsilon))$$

Then the posterior rate of convergence relative to the L_1 metric is $\epsilon_n = n^{-1/2} \log n$.

Lemma 4 is basically a modification of theorem 3.1 in Scricciolo (2011) to our proposed repulsive mixture model. Lemma 5 gives sufficient conditions for π to satisfy condition C1 and C2 in lemma 4.

Lemma 5. *Let π be defined as (2.1) and h be defined as either (2.2) or (2.3), then π satisfies condition C2 in lemma 4. Furthermore, if for a positive constant n_1 the function ξ satisfies $\xi(|x| \geqslant t) \lesssim \exp(-n_1 t^2)$, π satisfies condition C1 in lemma 4.*

As motivated above, when the number of mixture components is chosen to be conservatively large, it is appealing for the posterior distribution of the weights of the extra components to be concentrated near zero. Theorem 6 formalizes the rate of concentration with increasing sample size n. One of the main assumptions required in theorem 6 is that the posterior rate of convergence relative to the L_1 metric is $\delta_n = n^{-1/2}(\log n)^q$ with $q \geqslant 0$. We provided the contraction rate, under the proposed prior specification and univariate Gaussian kernel, in lemma 4. However, theorem 6 is a more general statement and it applies to multivariate mixture density of any kernel.

Theorem 6. *Let assumptions $B0 - B5$ be satisfied. Let π be defined as (2.1) and h be defined as either (2.2) or (2.3). If $\bar{\alpha} = \max(\alpha_1, \ldots, \alpha_k) < m/2$ and for positive constants r_1, r_2, r_3 the function g satisfies $g(x) \leqslant r_1 x^{r_2}$ for $0 \leqslant x < r_3$ then*

$$\lim_{M \to \infty} \limsup_{n \to \infty} E_n^0 \left[P \left\{ \min_{\{\iota \in S_k\}} \left(\sum_{i=k_0+1}^{k} p_{\iota(i)} \right) > M n^{-1/2} (\log n)^{q(1+s(k_0,\alpha)/s_{r_2})} \right\} \right] = 0$$

with $s(k_0, \alpha) = k_0 - 1 + mk_0 + \bar{\alpha}(k - k_0)$, $s_{r_2} = r_2 + m/2 - \bar{\alpha}$ *and* S_k *the set of all possible permutations of* $\{1, \ldots, k\}$.

Theorem 6 is a modification of theorem 1 in Rousseau and Mengersen (2011) to our proposed repulsive mixture model. Theorem 6 implies that the posterior expectation of weights of the extra components is of order $O(n^{-1/2}(\log n)^{q(1+s(k_0,\alpha)/s_{r_2})})$. When g is defined as (2.4), parameters r_1 and r_2 can be chosen such that $r_1 = \tau$ and $r_2 = \nu$.

When the number of components is unknown, with only an upper bound known, the posterior rate of convergence is equivalent to the parametric rate $n^{-1/2}$ (Ishwaran et al., 2001). In this case, the rate in theorem 6 is $n^{-1/2}$ under usual priors or our repulsive prior. However, in our experience using usual priors, the sum of the extra components can be substantial in small to moderate sample sizes, and often has high variability. As we show in Section 2.3, for repulsive priors the sum of the extra component weights is close to zero and has small variance for small as well as large sample sizes. When an upper bound on the number of components is unknown, the posterior rate of concentration is $n^{-1/2}(\log n)^q$ with $q > 0$. In this case, according to theorem 6, using our prior specification the logarithmic factor in theorem 1 of Rousseau and Mengersen (2011) can be improved.

2.2 Posterior Computation and Parameter Calibration

2.2.1 Posterior Computation

For posterior computation, we use a slice sampling algorithm (Neal, 2003), a class of Markov chain Monte Carlo algorithms widely used for posterior inference in infinite mixture models (Kalli et al., 2011). Letting g_0 be a conjugate prior, introduce a

latent variable u which is jointly modeled with γ through

$$\pi(\gamma_1, \ldots, \gamma_k, u) \propto \left(\prod_{h=1}^{k} g_0(\gamma_h) \right) 1\left\{ h(\gamma_1, \ldots, \gamma_k) > u \right\}.$$

Here $1(B)$ is the indicator function, equalling 1 if the event B occurs and 0 otherwise. Marginalizing out u, we recover the original density $\pi(\gamma_1, \ldots, \gamma_k)$. For a repulsion function defined as (2.3), let $B_j \equiv \bigcap_{\{s:s \neq j\}} [\gamma_j : g\{d(\gamma_s, \gamma_j)\} > u]$. When the repulsion function is defined as (2.2), one can introduce a latent variable for each product term. Under repulsive priors satisfying definition 1(i), the set B_j might not be easy to compute. However, when covariance matrices are constrained to be diagonal, vectors γ_js can be easily sampled element-wise. For multivariate observations, the location parameter vector can be sampled element-wise from truncated distributions.

For simplicity, assume that h is defined as 2.3, ψ is the Inverse-Gamma density with parameters (a_σ, b_σ), g_0 is the m-variate standard normal density and ϕ is the m-variate spherical normal kernel. Let $S_i \in \{1, \ldots, k\}$ be the variable indicating which cluster the ith observation belongs to. Let n_j be the number of data points in the jth cluster and let $\bar{y}_j$ be the average of observations in the jth cluster. Let $u^* = g^{-1}(u)$, $\alpha_p = (\alpha_1 + n_1, \ldots, \alpha_k + n_k)$ and $\gamma_j = (\mu_j, \sigma_j)$. Then the sampling algorithm can be summarized by the following steps:

Step 1. Update S_i, for $i \in \{1, \ldots, n\}$, by multinomial sampling

$$(S_i|-) \sim Mult(l_1, \ldots, l_k), \quad l_j = \frac{p_j \phi(y_i; \mu_j, \sigma_j I)}{\sum_{h=1}^{k} p_h \phi(y_i; \mu_h, \sigma_h I)};$$

Step 2. For repulsive priors satisfying definition 1(ii), sample (μ_j, σ_j) from

$$(\mu_j|-) \sim f_{\mu_j} \propto \mathcal{N}\left\{ (1 + n_j/\sigma_j)^{-1} \bar{y}_j n_j/\sigma_j, I(1 + n_j/\sigma_j)^{-1} \right\} 1\{\mu_j \in A(\mu_j)\}$$

$$(\sigma_j|-) \sim f_{\sigma_j} = \mathcal{IG}\left\{a_\sigma + \frac{n_j m}{2}, b_\sigma + \frac{1}{2}\sum_{\{i:S_i=j\}}(y_i-\mu_j)^T(y_i-\mu_j)\right\}$$

For repulsive priors satisfying definition 1(i) sample (μ_j, σ_j) from

$$\mu_j \sim f_{\mu_j} \quad ; \quad 1/\sigma_j \sim f'_{\sigma_j} \propto f_{\sigma_j} 1\{\sigma_j \in A(\sigma_j)\}$$

The set A is defined as $A(\cdot) = \{\cdot : d(\gamma_j, \gamma_s) > u^*, \forall s \neq j\}$ with $d(\cdot,\cdot)$ being defined as the symmetric K-L divergence for repulsive priors satisfying definition 1(i) and the Euclidean distance for repulsive priors satisfying definition 1(ii).

Step 3. Sample u and p from

$$(u|-) \sim Un\left\{0, h(\gamma)\right\}, \quad p \sim Dirichlet\left(\alpha_p\right)$$

2.2.2 Calibration

An important issue in implementing repulsive mixture models is elicitation of the repulsion hyper-parameters (τ, ν). Although a variety of strategies can be considered, we propose a simple approach that can be used to obtain a default hyper-parameter choice in general applications. In case (i) we choose $d(\cdot,\cdot)$ as the symmetric Kullback-Leibler divergence defined for Gaussian kernels as

$$s_{12} = d(\gamma_1, \gamma_2) = tr(\Sigma_1\Sigma_2^{-1}) + tr(\Sigma_1^{-1}\Sigma_2) - 2m + (\mu_1-\mu_2)^T(\Sigma_1^{-1}+\Sigma_2^{-1})(\mu_1-\mu_2),$$

while in case (ii) we use the Euclidean distance between the location parameters. For both case (i) and case (ii), define $\bar{d}$ as the mean of pairwise distances between atoms, $\bar{d} = \frac{1}{n(A)}\sum_{(s,j)\in A} d(\gamma_s, \gamma_j)$ with $A = \{(s,j) : s = 1, \ldots, k; j < s\}$ and $n(A)$ the cardinality of set A. Let f_1 and f_2 denote the densities of $\bar{d}$ under repulsive and non-repulsive priors respectively, with $(\varrho_j,\ \varsigma_j)$ the mean and standard deviation of f_j for $j = 1, 2$. We choose (τ, ν) so that f_1 and f_2 are well-separated using the following definition of separation (Dasgupta, 1999).

Def 7. *Given a positive constant c, f_1 and f_2 are c-separated if* $\varrho_1 - \varrho_2 \geqslant c \max(\varsigma_1, \varsigma_2)$.

We have found that $\nu = 2$ and $\nu = 1$ provide good default values in case (i) and (ii) respectively and we fix ν at these values in all our applications below. For a given value of ν, τ is found by starting with small values, estimating the mean and variance of $\bar{d}$ through Monte Carlo draws, and incrementing τ until definition 7 is satisfied for a pre-specified c. We use $c = 4$ in our implementations. A sensitivity analysis for different values of c can be found in appendix B.

2.3 Synthetic Examples

Simulation examples were considered to assess the performance of the repulsive prior in density estimation, clustering and emptying of extra components. Figure 2.2 plots the true densities in the various cases that we considered. For each synthetic dataset, repulsive and non-repulsive mixture models were compared considering a fixed upper bound on the number of components; extra components should be assigned small probabilities and hence effectively excluded. The slice sampler was run for $10,000$ iterations with a burn-in of $5,000$. The chain was thinned by keeping every 10th draw. To overcome the label switching problem, the samples were post-processed following the algorithm of Stephens (2000b). Details on parameters involved in the true densities, choice of prior distributions and methods used to compute quantities presented in this section can be found in Appendix B.

Repulsive mixtures satisfying definition 1(i) and non-repulsive mixtures were compared. For this experiment $1,000$ draws from a standard normal density and a two component mixture of overlapping normals was considered. Both repulsive and non-repulsive mixtures were run considering six as the upper bound of the number of components. Table 2.1 shows posterior summaries of parameters involved in the components with highest weights. Clearly, repulsive mixtures lead to a more parsi-

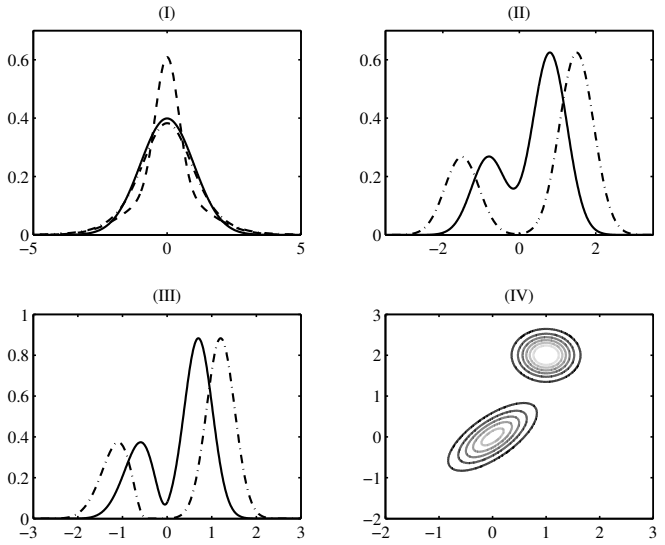

FIGURE 2.2: (I) Standard normal density (solid), two-component mixture of normals sharing the same location parameter (dash) and Student's t density (dash-dot), referred as (Ia, Ib, Ic), (II) two-components mixture of poorly (solid) and well separated (dot-dash) Gaussian densities, referred as (IIa, IIb), (III) mixture of poorly (solid) and well separated (dot-dash) Gaussian and Pearson densities, referred as $(IIIa, IIIb)$, (IV) two-components mixture of two-dimensional non-spherical Gaussians

monious representation of the true densities and more accurate parameter estimates. The mean and standard deviation of the K-L divergence under the first data example were $(0.003, 0.002)$ and $(0.004, 0.002)$ for non-repulsive and repulsive mixtures respectively; while under the second data example were $(0.006, 0.003)$ and $(0.009, 0.003)$ for non-repulsive and repulsive mixtures respectively. Therefore, repulsive mixtures were able to concentrate more on the reduced model while performing similarly to non-repulsive mixtures in estimating the true density.

Repulsive mixtures satisfying definition 1 (ii) and non-repulsive mixtures were compared to assess clustering performance. Table 2.2 shows summary statistics of the K-L divergence, the misclassification error and the sum of extra weights under repulsive and non-repulsive mixtures with six mixture components as the upper bound.

Table 2.2 shows also the misclassification error resulting from hierarchical clustering (Locarek-Junge and Weihs, 2009). In practice, observations drawn from the same mixture component were considered as belonging to the same category and for each dataset a similarity matrix was constructed. The misclassification error was established in terms of divergence between the true similarity matrix and the posterior similarity matrix. As shown in table 2.2, the K-L divergences under repulsive and non-repulsive mixtures become more similar as the sample size increases. For smaller sample sizes, the results are more similar when components are very well separated. Since a repulsive prior tends to discourage overlapping mixture components, a repulsive model might not estimate the density quite as accurately when a mixture of closely overlapping components is needed. However, as the sample size increases, the fitted density approaches the true density regardless of the degree of closeness among clusters. Again, though repulsive and non-repulsive mixtures perform similarly in estimating the true density, repulsive mixtures place considerably less probability on extra components leading to more interpretable clusters. In terms of misclassification error, the repulsive model outperforms the other two approaches while, in most cases, the worst performance was obtained by the non-repulsive model.

Potentially, one may favor fewer clusters, and hence possibly better separated clusters, by penalizing the introduction of new clusters more through modifying the precision in the Dirichlet prior for the weights; in appendix B, we demonstrate that this cannot solve the problem.

2.4 Real Data

We tested the performance of our proposed prior specification on three real datasets. The first involves 82 measurements of the velocities in km/s of galaxies diverging

Table 2.1: Posterior mean and standard deviation of weights, location and scale parameters under dataset drawn from densities (Ia, Ib)

	Density Ia			Density Ib					
	Comp 1			Comp 1			Comp 2		
	$\hat{p}_1$	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{p}_1$	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{p}_2$	$\hat{\mu}_2$	$\hat{\sigma}_2$
True	1	0	1	0.7	0	0.2	0.3	0	2
N-R	0.53	−0.01	0.85	0.44	0.08	1.21	0.34	0.12	1.33
	(0.16)	(0.04)	(0.25)	(0.06)	(0.10)	(1.05)	(0.06)	(0.16)	(1.11)
R	0.87	−0.00	0.84	0.67	−0.02	0.28	0.27	0.09	2.36
	(0.07)	(0.01)	(0.04)	(0.05)	(0.03)	(0.02)	(0.09)	(0.23)	(0.75)

from our own (Escobar and West (1995), Richardson and Green (1997)), the second consists of the acidity index measured in a sample of 155 lakes in north central Wisconsin (Richardson and Green (1997)), and the third consists of 150 observations from three different species of iris each with four measurements (Wang, 2010).

For the first two datasets, a repulsive mixture satisfying definition 1(i) was considered and a five-component mixture model was fit while for the third dataset a repulsive mixture satisfying definition 1(ii) was considered and both six components and ten components were considered as the upper bound. The same prior specification, Markov chain Monte Carlo sampler, and relabeling technique as in section 2.3 were utilized.

For the galaxy data, figure 2.3 reveals that there are three non-overlapping clusters with the one close to the origin relatively large compared to the others. Although this large cluster might be interpreted as two highly overlapping clusters, it appears to be well approximated by a single normal density. Richardson and Green (1997) and Escobar and West (1995) estimated the number of components, obtaining a posterior distribution on k concentrating on values ranging from 5 to 7. This may be due to the non-repulsive prior allowing closely overlapping components, favoring relatively large values of k. Figure 2.3 reveals that the non-repulsive prior specification leads to two overlapping and essentially indistinguishable clusters. Under repulsive

Table 2.2: Mean and standard deviation of K-L divergence, misclassification error and sum of extra weights resulting from non-repulsive mixture and repulsive mixture with a maximum number of clusters equal to six under different synthetic data scenarios.

		Ic	*IIa*	*IIb*	*IIIa*	*IIIb*	*IV*
n=100	K-L divergence						
	N-R	0.05	0.03	0.07	0.05	0.08	0.22
		(0.03)	(0.01)	(0.02)	(0.02)	(0.03)	(0.05)
	R	0.06	0.05	0.08	0.07	0.09	0.28
		(0.03)	(0.02)	(0.03)	(0.03)	(0.03)	(0.04)
	Misclassification						
	HCT	0.12	0.11	0.04	0.12	0.08	0.21
	N-R	0.69	0.26	0.06	0.17	0.05	0.13
		(0.10)	(0.10)	(0.04)	(0.09)	(0.06)	(0.05)
	R	0.53	0.18	0.01	0.10	0.01	0.05
		(0.10)	(0.09)	(0.02)	(0.05)	(0.01)	(0.02)
	Sum of extra weights						
	N-R	0.30	0.21	0.09	0.16	0.07	0.13
		(0.10)	(0.11)	(0.07)	(0.09)	(0.07)	(0.08)
	R	0.08	0.08	0.02	0.04	0.02	0.06
		(0.05)	(0.07)	(0.02)	(0.05)	(0.02)	(0.03)
n=1,000	K-L divergence						
	N-R	0.01	0.01	0.01	0.01	0.01	0.02
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
	R	0.01	0.01	0.01	0.01	0.01	0.03
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
	Misclassification						
	HCT	0.05	0.42	0.01	0.42	0.01	0.20
	N-R	0.65	0.24	0.03	0.14	0.03	0.19
		(0.11)	(0.08)	(0.04)	(0.09)	(0.03)	(0.02)
	R	0.46	0.13	0.00	0.03	0.00	0.17
		(0.16)	(0.04)	(0.01)	(0.02)	(0.01)	(0.01)
	Sum of extra weights						
	N-R	0.30	0.21	0.03	0.16	0.03	0.29
		(0.11)	(0.11)	(0.04)	(0.10)	(0.03)	(0.03)
	R	0.10	0.09	0.00	0.01	0.00	0.25
		(0.04)	(0.06)	(0.00)	(0.01)	(0.00)	(0.03)

priors, no clusters overlap significantly and unnecessary components receive a weight close to zero.

For the acidity data, figure 2.3 suggests that two clusters are involved. Since one of them appears to be highly skewed, we expect that three clusters might be needed to approximate this density well. Richardson and Green (1997) obtained a posterior for k almost equally concentrated on values of k ranging from 3 to 5. Figure 2.3 shows the estimated clusters for both repulsive and non-repulsive priors. With non-repulsive priors, four clusters receive significant weight and two of them overlap significantly. With repulsive priors, only three clusters receive significant weight and all of them appear fairly separated.

The iris data were previously analyzed by Sugar and James (2003) and Wang (2010) using new methods to estimate the number of clusters based on minimizing loss functions. They concluded the optimal number of clusters was two. This result did not agree with the number of species due to low separation in the data between two of the species. Such point estimates of the number of clusters do not provide a characterization of uncertainty in clustering in contrast to Bayesian approaches. Repulsive and non-repulsive mixtures were fitted under different choices of upper bound on the number of components. Since the data contains three true biological clusters, with two of these having similar distributions of the available features, we would expect the posterior to concentrate on two or three components. Posterior means and standard deviations of the three highest weights were $(0.30, 0.23, 0.13)$ and $(0.05, 0.04, 0.04)$ for non-repulsive and $(0.56, 0.29, 0.08)$ and $(0.05, 0.04, 0.03)$ for repulsive. Clearly, repulsive priors lead to a posterior more concentrated on two components, and assign low probability to more than three components. Figure 2.4 shows the density of the total probability assigned to the extra components. This quantity was computed considering the number of species as the true number of clusters. According to figure 2.4, our repulsive prior specification leads to extra

component weights very close to zero regardless of the upper bound on the number of components. The posterior uncertainty is also small. Non-repulsive mixtures assign large weight to extra components, with posterior uncertainty increasing considerably as the number of components increases.

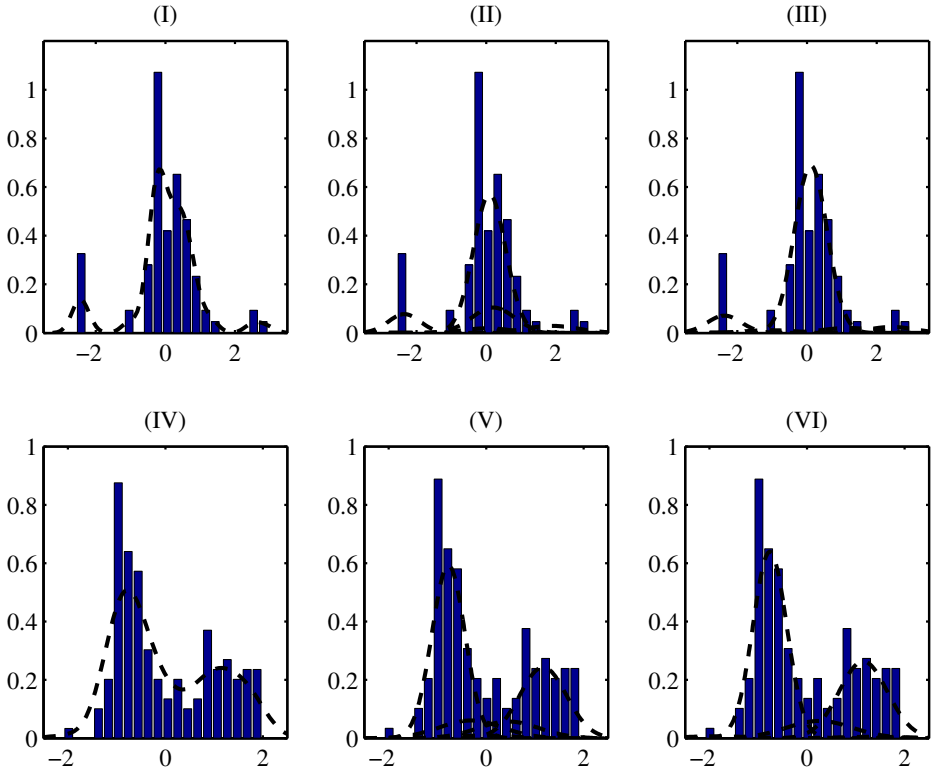

FIGURE 2.3: Histogram of galaxy data (I) and acidity data (IV) overlaid with a nonparametric density estimate using Gaussian kernel density estimation. Estimated clusters under galaxy data for non-repulsive (II) and repulsive (III) priors and under acidity data for non-repulsive (V) and repulsive (VI) priors

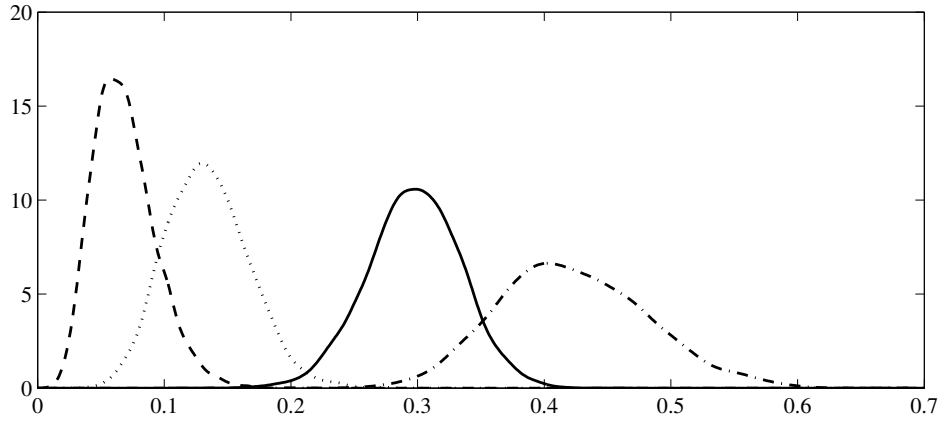

FIGURE 2.4: Density of sum of extra weights under k=6 for non-repulsive (solid) and repulsive (dash) and k=10 components for non-repulsive (dash-dot) and repulsive (dot)

3

Dictionary Learning for Conditional Distributions

Estimation of the conditional distribution of a response given high-dimensional features is a challenging problem arising in a number of important application areas, including neuroscience, genetics, and video processing. For example, one might desire automated estimation of a predictive density for a continuous or categorical neurologic phenotype of interest, such as intelligence or a creativity score, on the basis of available data for a patient including neuroimaging. The challenge is to estimate the probability density function of the phenotype based on a high-dimensional image of the subject's brain. In real data applications, often the relationship between predictors and response is non linear so that it is important to allow not only the mean but also the variance and shape of the response density to change flexibly with features, which are massive dimensional. In this chapter, we propose a novel stick breaking multiresolution that can flexibly and efficiently characterize the density of a response variable given high dimensional predictors. The algorithm scales efficiently to massive numbers of features, and can be implemented with slice sampling.

3.1 Methodology

We aim to build a flexible and scalable model for the density of $y \in \Re$ given a set of predictors. Let $x \in \mathcal{X} \subseteq \Re^p$ be a p-dimensional Euclidean vector-valued predictor random variable. Let $f(x)$ denote the marginal probability density of x. We assume that $f(x)$ concentrates around a lower-dimensional, possibly nonlinear, subspace $\mathcal{M}$. For example, $\mathcal{M}$ could be a union of affine subspaces, or a smooth compact Riemannian manifold. Let $y \in \mathcal{Y} \subseteq \Re$ be a real-valued target variable. Let x and y be sampled from some true but unknown joint distribution. We would like to learn $f(y|x)$. We assume that we obtain n independently and identically sampled observations, $(y_i, x_i^T)^T$ for $i \in \{1, \ldots, n\}$.

3.1.1 Model Overview

We propose a general modular approach to learn the conditional distribution of y given an high dimensional vector of predictors x consisting in two components: (i) a tree decomposition of the feature space, (ii) an assumed form of the conditional probability model. A tree decomposition $\mathcal{T}$ yields a multiscale partition of the data or the ambient space in which the data live. Starting from the coarsest scale, corresponding to the entire set, each set is split into two or more mutually exclusive subsets. This process continues until some convergence criteria is satisfied, e.g. the number of observations allocated to the finest scales is below some chosen threshold. Figure 3.1(i) shows a dyadic partition of the predictor space where a generic set $\mathcal{C}_{j,s}$ is partitioned into two subsets $\mathcal{C}_{j+1,s'}$ and $\mathcal{C}_{j+1,s''}$ such that

$$\mathcal{C}_{j,s} = \mathcal{C}_{j+1,s'} \bigcup \mathcal{C}_{j+1,s''} \quad , \quad \mathcal{C}_{j+1,s'} \bigcap \mathcal{C}_{j+1,s''} = \varnothing$$

For each scale j, the set of cells $C_j = \{C_{j,s}\}_{s=1}^{\mathcal{K}_j}$ provides a partition of $\mathcal{X}$. We define $j = 0$ as the root node/cell. For each $j > 0$, each $C_{j,s}$ has a unique parent node

$C_{j-1,s'}$ containing $C_{j,s}$, and conversely, any $C_{j,s} \subseteq C_{j-1,s'}$ is called a child of $C_{j-1,s'}$. Let the set of ancestors and descendants of $C_{j,s}$ be respectively defined as:

$$\mathcal{A}_{j,s} = \{(j', s') : j' < j, C_{j,s} \subseteq C_{j',s'}\} \quad , \quad \mathcal{D}_{j,s} = \{(j', s') : j' > j, C_{j',s'} \subseteq C_{j,s}\} \quad (3.1)$$

Considering a given tree decomposition $\mathcal{T}$ of $\mathcal{X}$, each $x_i \in \mathcal{X}$ has an associated path characterized by the sets including x_i (see figure 3.1(iii)). We assume that the density of y_i depends on x_i through this tree partition. Specifically, for each node (j, s) in the partition tree, we define a weight $\pi_{j,s}$ and dictionary density $f_{j,s}$ as shown in figure 3.1(ii). Then, the conditional density $f(y_i|x_i)$ will be a mixture of densities with components depending on the sets contained in the path of x_i (see figure 3.1(iv)). In the extreme case in which two predictor values x and x' belong to the same leaf partition sets, the conditional distributions $f(y'|x')$ and $f(y|x)$ will be identical. If the two paths differ only in the final generation or two, the conditional densities will typically be similar but not identical.

3.1.2 Model Specification

Assuming that the number of levels in the partition tree is k, we define the conditional density $f(y|x)$ as the convex combination of densities $\{f_{j,s_j(x)}\}_{j=1}^k$ with weights $\{\pi_{j,s_j(x)}\}_{j=1}^k$, i.e.

$$f(y|x) = \sum_{j=1}^{k} \pi_{j,s_j(x)} f_{j,s_j(x)}, \quad (3.2)$$

with $s_j(x)$ being the subset located at level j containing x, $(\pi_{j,s_j(x)}, f_{j,s_j(x)})$ being the weight and dictionary density associated to node $(j, s_j(x))$, $0 \leqslant \pi_{j,s_j(x)}$ and $\sum_{j=1}^k \pi_{j,s_j(x)} = 1$. According to model 3.2, only observations with predictors allocated to node (j, s), i.e. $\{y_i : x_i \in \mathcal{C}_{j,s}\}$, will have a mixture components with weight $\pi_{j,s}$

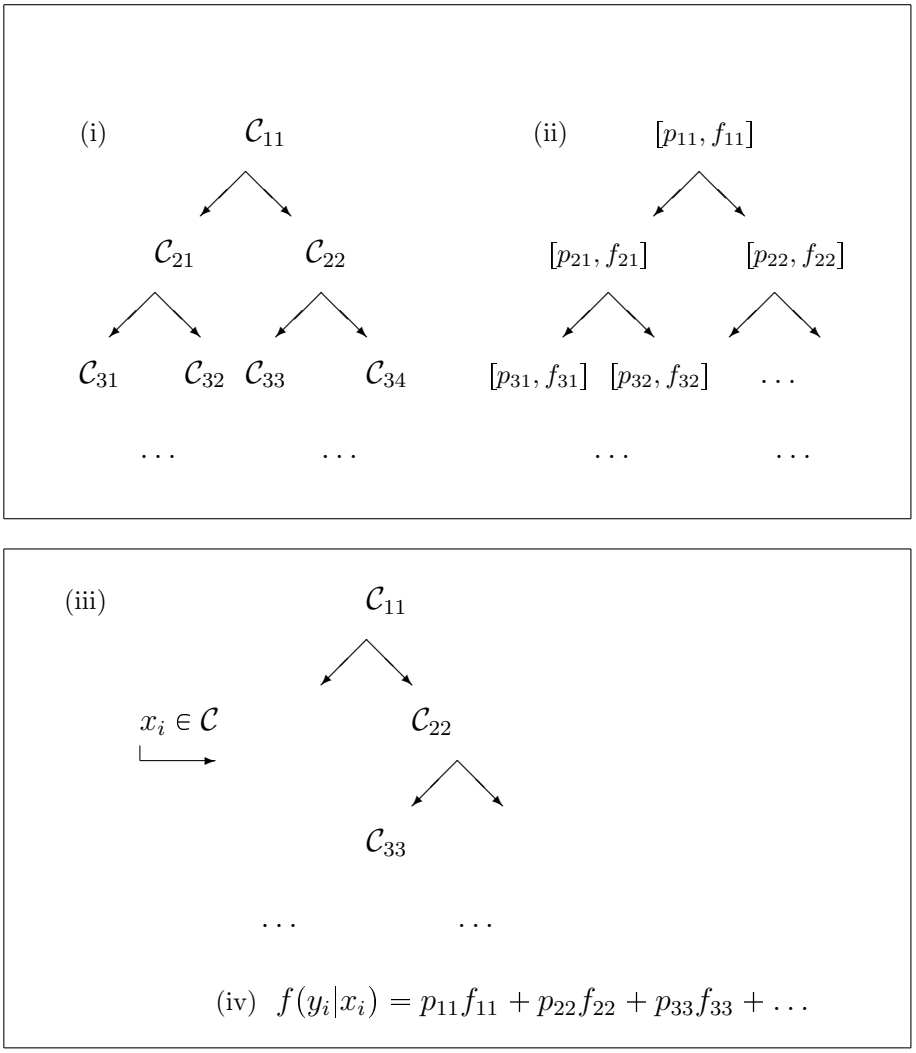

FIGURE 3.1: Partition tree schematic: (i) Multiscale partition of the data. (ii) Estimate dictionary density and weight associated to each set. (iii) Nodes along the tree containing $x_i \in \Re^p$. (iv) Conditional density of y_i given x_i defined as a convex combination of densities associated to the nodes containing x_i.

and density $f_{j,s}$. Notice that, as the weight associated to the first level of resolution approaches one, a non predictor-dependent density for y is obtained.

According to model (3.2), one observation can lie in subsets located at different resolution levels. This is critical in achieving a good compromise between bias and variance through borrowing information across different resolution levels. Though the proposed approach is reminiscent of a mixture of experts model (Jacobs et al., 1991), the two approaches are quite different, since under (3.2), neither mixture weights nor dictionary densities directly depend on predictors. This allows our model to scale efficiently to high dimensional predictors.

We let weights in 3.2 be generated by a stick-breaking process (Sethuraman, 1994b). For each node (j, s) in the partition tree, we define a stick length $V_{j,s} \sim \text{Beta}(1, \alpha)$. Then, we define weights in 3.2 as follows

$$\pi_{j,s} = V_{j,s} \prod_{(j',s') \in \mathcal{A}_{j,s}} [1 - V_{j',s'}], \quad \text{for} \quad j < k$$

with $\mathcal{A}_{j,s}$ defined as in 3.1 and $V_{k,s} = 1$ for all $s \in \{1, \ldots, \mathcal{K}_k\}$. This condition will ensure that, $\sum_{j=1}^{k} \pi_{j,s_j} = 1$ for any path $s = \{s_1, \ldots, s_k\}$. We refer to this prior as a multiresolution stick-breaking process. The parameter α encodes the complexity of the model, with $\alpha = 0$ corresponding to the case in which $f(y|x) = f(y)$.

3.2 Estimation

We desire a strategy that estimates posteriors over all potential marginal distributions so as to automatically obtain estimates of uncertainty. Moreover, we would like a procedure with a few hyper-parameters as possible. These motivate using a fully Bayesian strategy. A fully Bayesian approach would construct a large number of partitions, and integrate over them to obtain our posteriors. However, such a fully Bayesian is computational intractable for the ultrahigh-dimensionality problems

that motivate this work ($p \in \mathcal{O}(10^6)$), so we adopt a hybrid strategy. This hybrid strategy is based on a two stage algorithm where first the observations are allocated to different subsets in a tree fashion using an efficient partitioning algorithm and then, considering the partition as fixed, a multiresolution stick-breaking process is estimated using Bayesian methods.

Specifically, we employ METIS (Karypis and Kumar, 1999), a well-known relatively efficient graph partitioning algorithm with demonstrably good empirical performance on a wide range of graphs. We construct a weighted graph as done for the construction of diffusion maps (Coifman and Lafon, 2006). In practice, we add an edge between each pairs (x_v, x_u) and assign to any such edge weight $\mathrm{e}^{-\rho_{uv}^2}$, where ρ_{uv} is a given metric. In all applications below, ρ_{uv} is defined as the Euclidean distance between predictors x_u and x_v. Starting from the coarse scale, subsets will be split using METIS until the number of observations in the subsets located at the finest scale will drop below some chosen threshold τ. More formally let us define the number of levels k as the one satisfying the following conditions

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} 1(x_i \in C_{j,s_j}) \geqslant \tau, && \forall C_{j,s_j} \text{ with } j \leqslant k \\ &\textstyle\sum_{i=1}^{n} 1(x_i \in C_{k+1,s_{k+1}}) < \tau, && \text{f.s. } s_{k+1} \in \{1, \ldots, \mathcal{K}_{k+1}\} \end{aligned}$$

where $1(\cdot)$ is the indicator function and n is the number of observations. The above conditions imply that each subset located at the finest scale will have at least τ observations. Once the tree is constructed, we define the conditional density of y_i as the mixture density in 3.2. Though more complicated densities can be considered, dictionary densities $f_{j,s}$ will be estimated by assuming a normal form, i.e. $f_{j,s} = \mathcal{N}(\mu_{j,s}, \sigma_{j,s})$. In particular, densities corresponding to a particular partition set will be estimated considering only observations belonging to that partition set. To be specific, for estimating density $f_{j,s}$, we use observations $\mathcal{Y}_{j,s} = \{y_i : x_i \in \mathcal{C}_{j,s}\}$.

3.2.1 Full Conditionals

Parameters involved in the dictionary densities can be estimated using either frequentist or Bayesian methods. Bayesian methods are appealing since they can avoid singularities associated with traditional maximum likelihood inference, the prior has an appealing role as a regularizer, and we can characterize uncertainty in dictionary learning through the resulting posterior. Hence, parameters involved in dictionary densities will be estimated through Bayesian methods and inference on stick breaking weights and dictionary density parameters will be carried out using the Gibbs sampler.

For this purpose, introduce the latent variable $S_i \in \{1, \ldots, k\}$, for $i \in \{1, \ldots, n\}$, denoting the multiscale level used by the ith subject. Assuming data are normalized prior to analysis, we let $\mu_{j,s} \sim \mathcal{N}(0, I)$ and $\sigma_{j,s} = \mathcal{IG}(a, b)$ for the means and variances of the dictionary densities associated to node (j, s). Let $n_{j,s}$ be the number of observations allocated to node (j, s), i.e. $n_{j,s} = \sum_{i=1}^{n} 1(x_i \in C_{j,s}) 1(S_i = j)$. Define $\mathcal{I}_{j,s} = \{i : x_i \in \mathcal{C}_{j,s}\,,\, S_i = j\}$. Each Gibbs sampler iteration can be summarized in the following steps.

Step 1. Update S_i by sampling from the multinomial full conditional with

$$p(S_i = j \mid -) = \frac{\pi_{j,s_j(x_i)} f_{j,s_j(x_i)}(y_i)}{\sum_{h=1}^{k} \pi_{h,s_h(x_i)} f_{h,s_h(x_i)}(y_i)}$$

Step 2. Update $V_{j,s}$, for all $s \in \{1, \ldots, \mathcal{K}_j\}$ and $j \in \{1, \ldots, k\}$, by sampling from

$$p\left(V_{j,s}|-\right) = \text{Beta}(\beta_p, \alpha_p), \qquad \beta_p = 1 + n_{j,s} \qquad \alpha_p = \alpha + \sum_{\ell \in \mathcal{D}(j,s)} n_\ell$$

Step 3. Update $\mu_{j,s}$, for all $s \in \{1, \ldots, \mathcal{K}_j\}$ and $j \in \{1, \ldots, k\}$, by sampling from

$$p(\mu_{j,s}|-) = \mathcal{N}\left(M_{j,s} n_{j,s} / \sigma_{j,s} \bar{y}_{j,s}, M_{j,s}\right)$$

with $M_{j,s} = (1 + n_{j,s}/\sigma_{j,s})^{-1}$, $\bar{y}_{j,s} = \sum_{i \in \mathcal{I}_{j,s}} y_i$.

Step 4. Update $\sigma_{j,s}$, for all $s \in \{1, \ldots, \mathcal{K}_j\}$ and $j \in \{1, \ldots, k\}$, by sampling from

$$p(\sigma_{j,s}|-) = \mathcal{IG}\left(a + n_{j,s}/2, b + \bar{s}/2\right)$$

where $\bar{s} = \sum_{i \in \mathcal{I}_{j,s}} (y_i - \mu_{j,s})^2$.

3.2.2 Predictions

Consider the case we want to predict the response y_{n+1} for a future observation based on predictors x_{n+1} and previous observations $(x^{(n)}, y^{(n)})$ with $x^{(n)} = (x_1, \ldots, x_n)$ and $y^{(n)} = (y_1, \ldots, y_n)$. Because the partitioning strategy that we adopted lacks an elegant out-of-sample embedding function (unlike other partitioning strategies), we adopt a Voronoi expansion procedure by which the new vector of features x_{n+1} is allocated to $\mathcal{C}_{j,k}$'s having the closest centers with respect to some metric ρ. Summaries of the predictive density of y_{n+1} will be computed as follows:

(i) for each scale $j \leqslant k$, allocate predictors x_{n+1} to $\mathcal{C}_{j,k}$'s having the closest centers with respect to ρ

(ii) run the Gibbs sampler for H iterations, and at the hth iteration:

a) sample parameters $\left\{\sigma_{j,s}^{(h)}, \mu_{j,s}^{(h)}, \pi_{j,s}^{(h)}\right\}_{\forall j, \forall s}$ from the posterior, following the procedure explained in §3.2.1

b) sample $\hat{y}_{n+1}^h$ from

$$\hat{y}_{n+1}^s \sim \sum_{j=1}^{k} \pi_{j,s(x_{n+1})}^{(h)} \mathcal{N}\left(\mu_{j,s(x_{n+1})}^{(h)}, \sigma_{j,s(x_{n+1})}^{(h)}\right)$$

(iii) given the sequence $\left\{\hat{y}_{n+1}^h\right\}_{h=1}^{H}$, summaries of the predictive density of the response variable such as mean, variance and quantiles can be computed.

3.3 Simulation Studies

In order to assess the predictive performance of the proposed model, different simulation scenarios were considered. Let n be the number of observations, $y \in \Re$ the response variable and $x \in \Re^p$ a set of predictors. The Gibbs sampler was run considering $20,000$ as the maximum number of iterations with a burn-in of $1,000$. Gibbs sampler chains were stopped testing normality of normalized averages of functions of the Markov chain (Chauveau and Diebolt, 1998). Parameters (a, b) and α involved in the prior density of parameters $\sigma_{j,s}$s and $V_{j,s}$s were set respectively equal to $(3, 1)$ and 1. The threshold τ used to determine the number of levels was set equal to 5. Let the metric utilized to allocate new predictors (defined as ρ in §3.2.2) be the Euclidean distance.

In all simulation scenarios, predictors were assumed to belong to an r-dimensional space, either a lower dimensional plane or a non linear manifold, with $r << p$. For each synthetic dataset, the proposed model was compared with CART and lasso in terms of mean squared error. For CART and Lasso standard Matlab packages were utilized and the regularization parameter of Lasso was chosen based on the AIC.

3.3.1 Illustrative Example

Consider

$$x_i \sim \mathcal{N}(\psi(\mu_i), \sigma^2 I),$$

where $\Psi = \{\psi \colon \mathcal{M} \to \Re^p\}$, $\mu_i \in \mathcal{M}$, $\sigma \in (0, \infty)$, I is the $p \times p$ dimensional identity matrix. Let $\mathcal{M}$ be a smooth compact Riemannian manifold. For simplicity, let us assume that $\mathcal{M}$ is a curve. Let $\psi(\mu) = 1\mu$ with 1 being a p-dimensional vector with all elements equal to 1. Define the conditional $f(y|x)$ as a function of μ, i.e. a mixture density with mixture weights depending on μ. We will show that our construction facilitates an estimate of the density of y.

Specifically, we created an equally spaced grid of points $t_i \in \{0, \ldots, 20\}$. Then, we let $\eta_i = \sin(t_i)$ and predictors be a linear function of η_i plus Gaussian noise, i.e. $x_{ij} = \eta_i + \epsilon_{ij}$ with $\epsilon_{ij} \sim N(0, 0.1)$ for $j \in \{1, \ldots, p\}$. In particular, we set $p = 1,000$. The response was drawn from the following mixture of Gaussians

$$y_i \sim w_i \mathcal{N}(-2, 1) + (1 - w_i)\mathcal{N}(2, 1) \tag{3.3}$$

with $w_i = |\eta_i|$. Figure 3.2 shows the estimated density of four data points. These estimates were obtained by performing leave-one-out prediction for different number of observations in the training set. As the figure clearly shows our construction facilitates an estimate of the density y approaching the true density as the number of observations in the training set increases.

3.3.2 Linear Lower Dimensional Space

In this section, the vector of predictors is assumed to lie close to a lower dimensional plane. In practice, predictors were modeled through a factor model as follows

$$x_i = \Lambda \eta_i + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}_p(0, \Sigma_0) \qquad \eta_i \sim \mathcal{N}_r(0, I) \tag{3.4}$$

with $\Sigma_0 = diag(\sigma_1, \ldots, \sigma_p)$, Λ being a $p \times r$ matrix and $r << p$. In the first simulation scenario the response y was assumed to be a function of the latent variable η so that the dependence between response and predictors was induced by the shared dependence on the latent factors. In practice, the vector $z_i = (y_i, x_i^T)^T$ was jointly sampled from a factor model. The loading matrix was derived as the product of a matrix with orthogonal columns and a diagonal matrix with positive elements on the diagonal, i.e. $\Lambda = \Gamma\Theta$. In particular, the columns of Γ were uniformly sampled from the Stiefel manifold while the diagonal matrix of Θ were sampled from an inverse Gamma with shape and rate parameters $(1, 4)$.

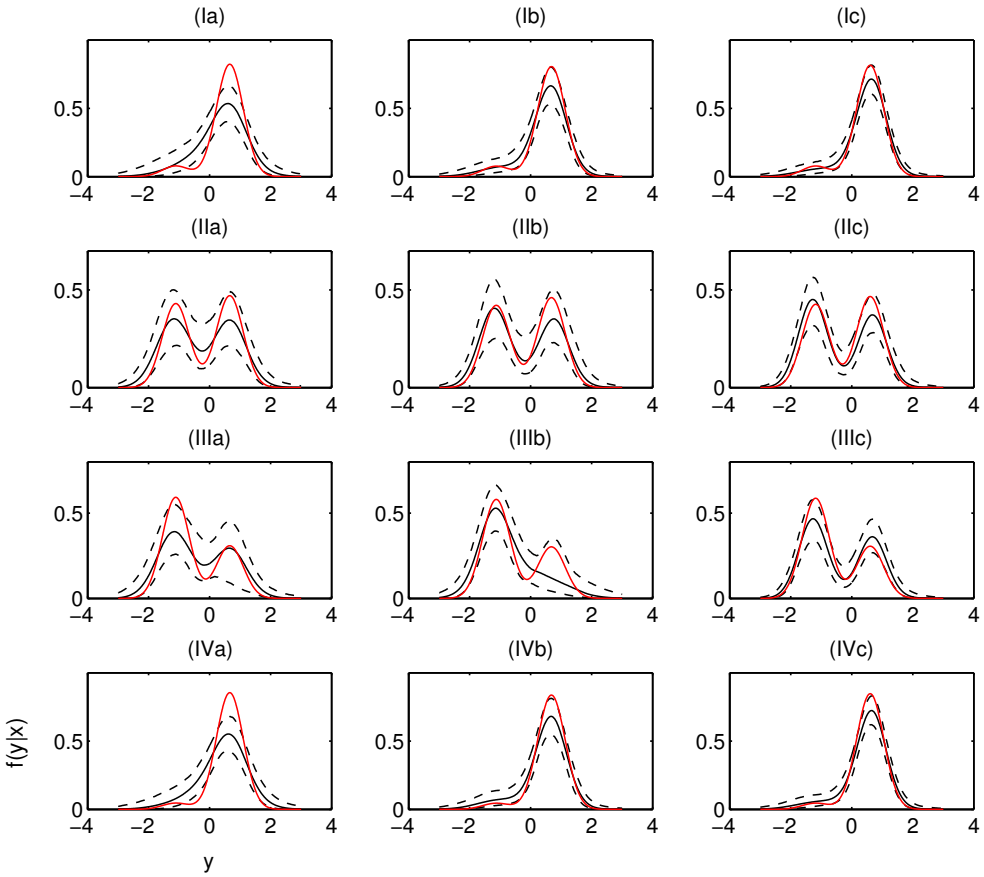

FIGURE 3.2: Illustrative example: Plot of true density (red line) and estimated density (50th percentile: solid line, 2.5th and 97.5th percentiles: dashed lines) for four data points (I, II, III, IV) considering different training set size (a:100, b:200, c:300).

In the second simulation scenario, x was sampled from the factor model above, while y was sampled from a normal with location and scale parameter $(1, 1)$ if the first variable was positive, i.e. $x_1 > 0$, and from a normal with location and scale $(-1, 1)$ otherwise. In both examples, an inverse Gamma prior with parameters $(1, 4)$ were utilized for σ_j with $j \in \{1, \ldots, p\}$.

3.3.3 Non-Linear Lower Dimensional Space

In this section predictors were assumed to lie close to a lower dimensional non-linear manifold. In the first simulation study, predictors and response were jointly sampled from an N components mixture of factor analyzers so that the vector of predictors and response were assumed to lie close to N lower dimensional planes. For each mixture components, the loading matrix and variances were sampled as in the first simulation scenario in §3.3.2, while mixture weights were sampled from a Dirichlet distribution with parameter $\alpha_j = 1$ for $j \in \{1, \ldots, N\}$. The number of latent factors was considered to be increasing in the number of components. In particular, we let the hth mixture component be modeled through h factors.

In the other two simulation scenarios predictors were assumed to lie close to the Swissroll and the S-manifold, all two dimensional manifold embedded in $\Re^p$, while the response was sampled from a normal with mean equal to one of the coordinates of the manifold and standard deviation one. Figure 3.3 shows the Swissroll and the S-manifold embedded in $\Re^3$.

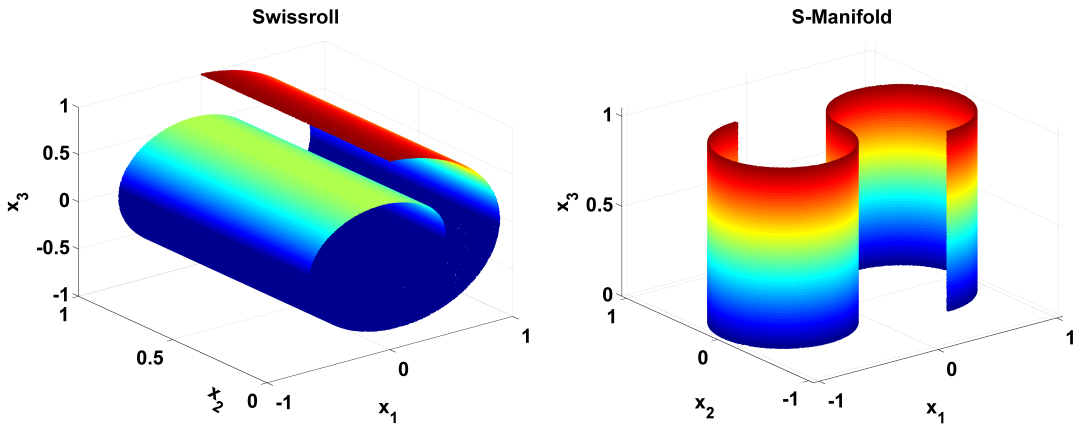

FIGURE 3.3: Swissroll-Manifold and S-Manifold embedded in $\mathcal{R}^3$

3.3.4 Results

For each simulation scenario in §3.3.2 and §3.3.3, we sampled $M = 20$ datasets involving up to 300 observations and for each method we performed leave-one-out predictions. Table 3.1, 3.2, 3.3 and 3.4 show mean squared errors under the proposed approach, CART and lasso for each data scenario. As shown, in almost all data scenarios, our model is able to perform as well as or better than the model associated to the lowest mean squared error. In particular, when the response is a non linear function of predictors, CART performs better than Lasso (table 3.2), while when a linear relationship is assumed lasso outperforms CART (table 3.1, 3.3, and 3.4). The tables also report the mean of CPU usage to predict a single point as a function of the number of features. In particular, CPU time is expressed in seconds and codes have been running on our workstation (Intel Core i7-2600K Quad-Core Processor memory 8192 MB). Clearly, the proposed model scale substantially better than others to high dimensional predictors.

Beside running simulations and reporting the distribution of performance for each algorithm, we compare the algorithms per simulation. Define $r_m^{\mathcal{W}}$ defined as

$$r_m^{\mathcal{W}} = \phi(MSB)/\phi(\mathcal{W}),$$

where ϕ is the quantity of interest (for example, CPU time in seconds or mean squared error), MSB is our approach and $\mathcal{W}$ is the competitor algorithm. To obtain mean-squared error estimates from MSB, we select our posterior mean as a point-estimate (the comparison algorithms do not generate posterior predictions, only point estimates). For each simulation scenario, we sampled multiple datasets and compute the *matched* distribution of $r_m^{\mathcal{W}}$. This provides a much more informative indication of algorithmic performance, in that we indicate the fraction of simulations one algorithm outperforms another on some metric. This is akin to power gained by matched two-sample tests. For each example, we sampled 20 datasets to obtain estimates of the

distribution over $r_m^{\mathcal{W}}$.

Figures 3.4 and 3.5 depict the relative mean-squared error and CPU time in seconds of our approach, versus CART (red) and Lasso (black) for different simulation scenarios. The three simulation scenarios are: linear subspaces, union of linear subspaces (MFA) and the swissroll. MSB outperforms both CART and Lasso in all three scenarios regardless of ambient dimension ($r_{mse}^{\mathcal{W}} < 1$ for all p).

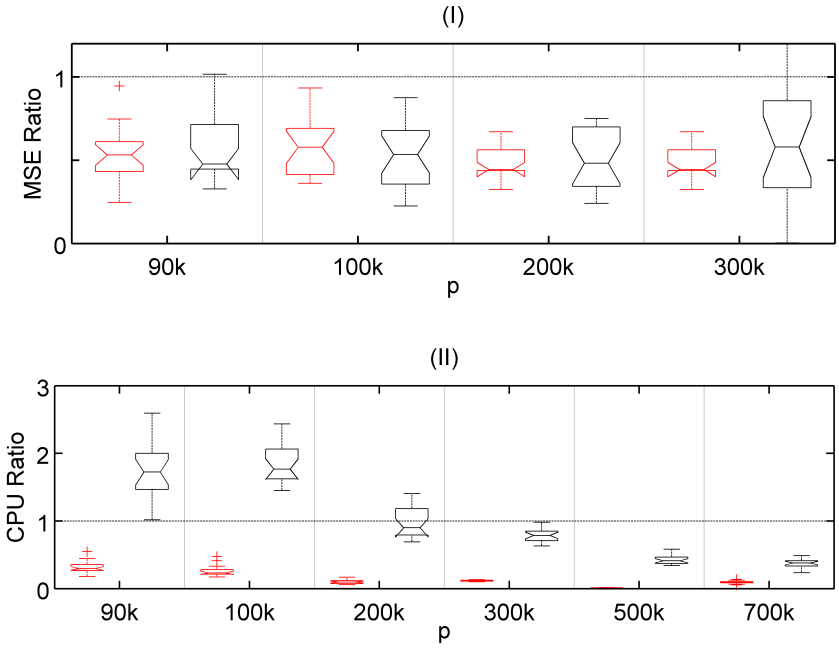

FIGURE 3.4: Numerical results for various simulation scenarios. Top plot depicts the relative mean-squared error of MSB (our approach), versus CART (red) and Lasso (black) as a function of ambient dimension of x. Bottom plot depicts the ratio of CPU time as a function of ambient dimension of x. The simulation scenario considered is the linear subspace. MSB outperforms both CART and Lasso regardless of ambient dimension ($r_{mse}^{\mathcal{W}} < 1$ for all p). MSB compute time is relatively constant as p increases, whereas Lasso's compute time increases, thus, as n or p increase, MSB CPU time becomes less than Lasso's. MSB was always significantly faster than CART and PC regression, regardless of n or p. For all panels, $n = 100$ when p varies, and $p = 300$k when n varies, where k indicates 1000, e.g., 300k$= 3 \times 10^5$.

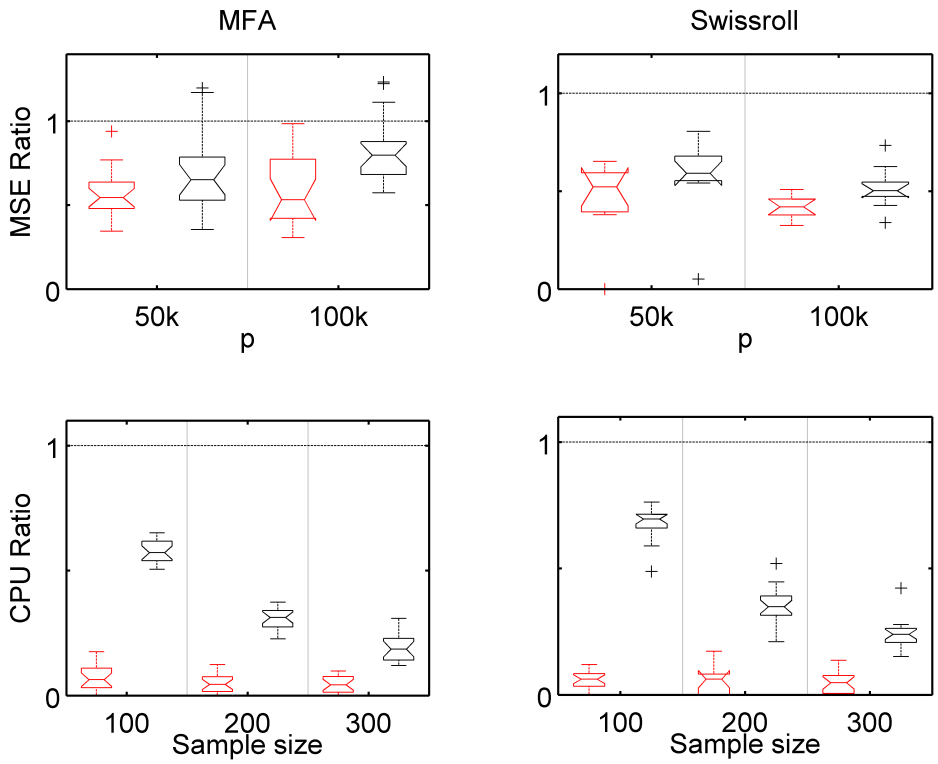

FIGURE 3.5: Numerical results for various simulation scenarios. Top plots depict the relative mean-squared error of MSB (our approach), versus CART (red) and Lasso (black) as a function of ambient dimension of x. Bottom plots depict the ratio of CPU time as a function of sample size. The two simulation scenarios are: MFA (left) and Swissroll (right). MSB outperforms both CART and Lasso in all two scenarios regardless of ambient dimension ($r^{\mathcal{W}}_{mse} < 1$ for all p). MSB compute time is relatively constant as n or p increase, whereas Lasso's compute time increases, thus, as n or p increase, MSB CPU time becomes less than Lasso's. MSB was always significantly faster than CART and PC regression, regardless of n or p. For all panels, $n = 100$ when p varies, and $p = 300$k when n varies, where k indicates 1000, e.g., 300k$= 3 \times 10^5$.

3.4 Real Application

We assessed the predictive performance of the proposed method on two very different neuroimaging datasets. First, we consider a structural connectome dataset collected at the Mind Research Network. Data were collected as described in Jung et al. (2010).

For the analysis, all variables were normalized by subtracting the mean and dividing by the standard deviation. The prior specification and Gibbs sampler described in §3.3 were utilized.

In the first experiment we investigated the extent to which we could predict creativity (as measured via the Composite Creativity Index (Arden et al., 2010)). For each subject, we estimate a 70 vertex undirected weighted brain-graph using the Magnetic Resonance Connectome Automated Pipeline (Gray et al., 2010) from diffusion tensor imaging data (Mori and Zhang, 2006). Because our graphs are undirected and lack self-loops, we have a total of $p = \binom{70}{2} = 2,415$ potential weighted edges. The p-dimensional feature vector consists of the natural logarithm of the weight for each edge.

The second dataset comes from a resting-state functional magnetic resonance experiment as part of the Autism Brain Imaging Data Exchange. We selected the Yale Child Study Center for analysis. Each brain-image was processed using the Configurable Pipeline for Analysis of Connectomes (Sikka et al., 2012). For each subject, we computed a measure of normalized power at each voxel called fALFF (Zou et al., 2008). To ensure the existence of nonlinear signal relating these predictors, we let y_i correspond to an estimate of overall head motion in the scanner, called mean framewise displacement (FD) computed as described in Power et al. (2012). In total, there were $p = 902,629$ voxels.

Table 3.5 shows mean and variance squared error based on leave-one-out predictions. For each data example, we report the mean and standard deviation (s.d.) across subjects of squared error, and CPU time (in seconds). For the first data example, we compared our approach (multiscale stick-breaking; MSB) to CART, lasso and random forests. Table 3.5 shows that MSB outperforms all the competitors in terms of mean square error; this is in addition to yielding an estimate of the entire conditional density for each y_i. It is also significantly faster that random forests, the

next closest competitor, and faster than lasso. For this relatively low-dimensional example, CART is reasonably fast. For the second data application, given the huge dimensionality of the predictor space, we were unable to get either CART or random forest to run to completion, yielding memory faults on our workstation (Intel Core i7-2600K Quad-Core Processor memory 8192 MB). We thus only compare performance to lasso. As in the previous example, MSB outperforms lasso in terms of predictive accuracy measured via mean-squared error, and significantly outperforms lasso in terms of computational time.

Table 3.1: Linear manifold example 1: Mean and standard deviations of squared errors under multiscale stick-breaking (MSB), CART and Lasso for sample size 50 and 100 for different simulation scenarios. Variable time indicates the mean of CPU usage to predict a single point, p is the dimensionality of the predictor space, n is the sample size and k indicates $1,000$, i.e. 300k=300,0000. **Bold** indicates best MSE, $*$ indicates best CPU time. As shown, MSB outperforms both CART and Lasso regardless of ambient dimension and sample size.

			$r = 5$			$r = 10$		
p	n		MSB	CART	LASSO	MSB	CART	LASSO
		MSE	**0.18**	0.31	0.25	**0.22**	0.58	0.22
10K	50		**(0.32)**	(0.30)	(0.42)	**(0.24)**	(0.54)	(0.30)
		TIME	**3**	2	1*	**3**	3	1*
		MSE	**0.18**	0.27	0.26	**0.20**	0.41	0.52
10K	100		**(0.26)**	(0.42)	(0.46)	**(0.23)**	(0.46)	(0.78)
		TIME	**5**	5	2*	**5**	5	1*
		MSE	**0.35**	0.45	0.89	**0.16**	0.33	0.20
100K	50		**(0.53)**	(0.77)	(1.04)	**(0.21)**	(0.46)	(0.31)
		TIME	**3**	25	2*	**13**	27	2*
		MSE	**0.43**	0.88	0.52	**0.17**	0.50	0.31
100K	100		**(0.59)**	(1.29)	(0.70)	**(0.24)**	(0.75)	(0.49)
		TIME	**7**	50	5*	**7**	51	5*
		MSE	**0.11**	0.16	0.15	**0.83**	2.26	0.92
500K	50		**(0.15)**	(0.24)	(0.19)	**(1.01)**	(2.60)	(3.69)
		TIME	**5***	90	11	**5***	121	10
		MSE	**0.003**	0.17	0.08	**0.13**	1.37	1.06
500K	100		**(0.16)**	(0.23)	(0.13)	**(1.12)**	(1.80)	(1.50)
		TIME	**10***	214	43	**8***	227	42
		MSE	1.70	1.48	**1.47**	**0.66**	1.65	1.07
700K	50		(2.18)	(2.47)	**(1.63)**	**(0.87)**	(1.49)	(0.95)
		TIME	6*	121	**12**	**7***	151	13
		MSE	**0.69**	1.36	0.82	**0.78**	1.52	1.43
500K	100		**(0.94)**	(1.47)	(1.28)	**(1.03)**	(1.34)	(2.11)
		TIME	**13***	321	41	**12***	325	44

Table 3.2: Linear manifold example 2: Mean and standard deviations of squared errors under multiscale stick-breaking (MSB), CART and Lasso for sample size 50 and 100. Variable time indicates the mean of CPU usage to predict a single point, p is the dimensionality of the predictor space, n is the sample size and k indicates $1,000$, i.e. 300k=300,0000. In this case, given the non-linear relationship between response and predictors, CART outperforms Lasso. However, our model results in the lowest mean squared errors.

			$r = 2$			$r = 5$		
p	n		MSB	CART	LASSO	MSB	CART	LASSO
10K	100	MSE	**1.54**	1.78	2.37	**0.84**	1.25	1.62
		STD	**(1.70)**	(1.72)	(0.89)	**(1.38)**	(1.35)	(1.47)
50K	100	MSE	**0.76**	0.97	1.77	**0.88**	1.53	1.43
		STD	**(1.04)**	(1.21)	(3.13)	(**1.00**)	(1.59)	(2.73)
100K	100	MSE	**0.77**	1.01	1.61	0.67	**0.46**	0.97
		STD	**(0.94)**	(1.13)	(1.85)	(0.82)	**(0.61)**	(1.16)
200K	100	MSE	**0.86**	0.90	1.41	**0.74**	1.09	0.78
		STD	**(1.30)**	(1.35)	(1.41)	**(0.95)**	(1.98)	(0.95)

Table 3.3: Non-linear manifold - MFA: Mean and standard deviations of squared errors under multiscale stick-breaking (MSB), CART and Lasso for sample size 50 and 100 for different simulations sampled from a mixture of factor analyzers. Variable time indicates the mean of CPU usage to predict a single point, p is the dimensionality of the predictor space, n is the sample size and k indicates $1,000$, i.e. 300k=300,0000. **Bold** indicates best MSE, $*$ indicates best CPU time. As shown, MSB outperforms both CART and Lasso regardless of ambient dimension and sample size.

			$N = 10$			$N = 5$		
p	n	SIM	MSB	CART	LASSO	MSB	CART	LASSO
		MSE	**0.23**	0.42	0.36	**0.17**	0.43	0.22
50K	100		**(0.34)**	(0.59)	(0.43)	**(0.18)**	(0.69)	(0.23)
		TIME	**5**	24	3*	**7**	27	3*
		MSE	**0.23**	0.42	0.27	**0.17**	0.22	0.20
50K	200		**(0.33)**	(0.56)	(0.23)	**(0.19)**	(0.38)	(0.25)
		TIME	**10**	51	8*	**12**	56	7*
		MSE	**0.67**	1.35	1.32	**0.15**	0.17	0.22
100K	100		**(1.04)**	(2.26)	(1.36)	**(0.23)**	(0.19)	(0.23)
		TIME	**9**	47	6*	**6**	44	5*
		MSE	**0.64**	1.37	0.85	**0.15**	0.26	0.15
100K	200		**(0.95)**	(1.77)	(1.29)	**(0.24)**	(0.42)	(0.24)
		TIME	**14***	99	15	**11***	89	15
		MSE	**0.26**	0.39	0.31	**0.63**	1.40	1.01
300K	100		**(0.39)**	(0.51)	(0.52)	**(0.80)**	(1.24)	(1.46)
		TIME	**9***	125	18	**9***	145	17
		MSE	**0.25**	0.47	0.26	**0.63**	1.17	0.92
300K	200		**(0.36)**	(0.88)	(0.43)	**(0.80)**	(2.11)	(1.04)
		TIME	**15***	262	40	**13***	283	43
		MSE	**0.25**	0.30	0.30	**0.62**	1.42	0.70
300K	300		**(0.36)**	(0.41)	(0.48)	**(0.89)**	(1.85)	(0.94)
		TIME	**15***	463	73	**16***	465	89

Table 3.4: Non-linear manifold - Swissroll and S-Manifold: Mean and standard deviations of squared errors under multiscale stick-breaking (MSB), CART and Lasso for sample size 50 and 100 for different simulation scenarios. Variable time indicates the mean of CPU usage to predict a single point, p is the dimensionality of the predictor space, n is the sample size and k indicates $1,000$, i.e. 300k=300,0000. **Bold** indicates best MSE, $*$ indicates best CPU time. As shown, MSB outperforms both CART and Lasso regardless of ambient dimension and sample size.

			Swissroll			S-Manifold		
p	n		MSB	CART	LASSO	MSB	CART	LASSO
		MSE	**0.24**	0.44	0.25	**0.38**	0.38	0.84
100k	50		**(0.24)**	(0.42)	(0.29)	**(0.40)**	(0.35)	(0.80)
		TIME	**3**	22	2*	**5**	7	1*
		MSE	0.24	0.43	**0.17**	**0.25**	0.30	0.70
100k	100		(0.26)	(0.55)	**(0.22)**	**(0.22)**	(0.25)	(0.50)
		TIME	6*	48	**7**	**7***	50	7
		MSE	**0.24**	0.67	0.29	**0.35**	0.40	0.73
200k	50		**(0.23)**	(0.50)	(0.29)	**(0.22)**	(0.30)	(0.40)
		TIME	**4***	38	5	**3***	40	5
		MSE	**0.25**	0.78	0.33	**0.37**	0.37	0.70
200k	100		**(0.26)**	(0.74)	(0.36)	**(0.25)**	(0.27)	(0.55)
		TIME	**6***	96	13	**6***	98	14
		MSE	**0.17**	0.47	0.23	**0.16**	0.20	0.35
500k	50		**(0.23)**	(0.43)	(0.22)	**(0.20)**	(0.19)	(0.40)
		TIME	**5***	126	10	**5***	130	15
		MSE	**0.17**	0.33	0.19	**0.11**	0.25	0.56
500k	100		**(0.21)**	(0.46)	(0.23)	**(0.14)**	(0.20)	(0.61)
		TIME	**11***	230	25	**10***	254	27

Table 3.5: Neuroscience application quantitative performance comparisons. Squared error predictive accuracy per subject (using leave-one-out) was computed. We report the mean and standard deviation (s.d.) across subjects of squared error, and CPU time (in seconds). We compare multiscale stick-breaking (MSB), CART, Lasso and random forest (RF). MSB outperforms all the competitors in terms of predictive accuracy and scalability. Only MSB and Lasso even ran for the $\approx 10^6$ dimensional application. **Bold** indicates best MSE, * indicates best CPU time.

DATA	n	p	MODEL	MSE (S.D.)	TIME (S.D.)
CREATIVITY	108	2,415	**MSB**	**0.56 (0.85)**	**1.1 (0.02)**
			CART	1.10 (1.00)	0.9 (0.01)
			LASSO*	0.63 (0.95)*	0.40 (0.10)*
			RF	0.57(0.90)	78.2 (0.59)
MOVEMENT	56	$\approx 10^6$	**MSB***	**0.76 (0.90)***	**20.98 (2.31)***
			LASSO	1.02 (0.98)	96.18 (9.66)

4

Bayesian factor trees

In this chapter, we focus on the problem of modeling the density of a vector of observations $y \in \mathcal{Y} \subseteq \Re^p$ given a set of predictors $x \in \mathcal{X} \subseteq \Re^q$. One natural approach to estimating conditional multivariate densities is to let $z = (y^T, x^T)^T$ and then model the combined vector as iid from an unknown density g, with the conditional density $f(y|x)$ obtained as a byproduct. This joint modeling trick has been widely used for univariate response (Muller et al., 1996; Shahbaba and Neal, 2009; Hannah et al., 2011), and is just as applicable in the multivariate case. To characterize the unknown density $g(z)$, one can use a mixture of multivariate Gaussian densities. However, as $p + q$ increases, the curse of dimensionality prevents one from obtaining adequate performance. This problem can be overcome by incorporating dimensionality reduction within each component by using a factor model, leading to a mixture of factor analyzers (MFA) (Tipping and Bishop, 1997). Under MFA the $(p + q)$-dimensional vector is modeled as

$$g(z) = \int \mathcal{N}_{p+q}\left(z|\mu, \Lambda\Lambda^T + \Sigma\right) d\mathcal{P}(\mu, \Lambda, \Sigma) \tag{4.1}$$

Model 4.1 can efficiently characterize the density of high-dimensional observations,

while conducting dimensionality reduction through learning the lower-dimensional data manifold using a piecewise planar approximation (Chen et al., 2010). Unfortunately, using the joint modeling trick to induce an estimate for $f(y|x)$ from an estimate for $g(z)$, with $z = (y^T, x^T)^T$, tends to have poor performance in practice, particularly when x is higher dimensional than y. One pays a heavy computational price for estimating the high-dimensional nuisance parameter corresponding to the marginal density of x, and additionally learning of a parsimonious low-dimensional structure tends to be driven largely by the x marginal, leading to relatively poor performance in estimating $f(y|x)$. One can alternatively use (4.1) for the conditional density $f(y|x)$ directly, with x dependence incorporated in the mixing measure or factor analytic parameters. This can be accomplished using previous nonparametric Bayes machinery (Ren et al., 2011; Rodriguez and Dunson, 2011; Hatjispyros et al., 2011). However, as p and q increase, computation rapidly becomes prohibitively slow. In this chapter, we propose a Bayesian factor tree model that can flexibly and efficiently learn the density of a p dimensional response given an high dimensional vector of features.

4.1 Methodology

4.1.1 Model Structure

We aim to reduce dimensionality for tractability in building a flexible and scalable model for the predictor-dependent density of $y \in \Re^p$. The density of y will depend on covariates through the multiscale representation of the data presented in §3.1.1 (refer to this section for further details). Given the multiscale partition, the conditional density $f(y|x)$ is defined as

$$f(y|x) = \sum_{j=1}^{k} \pi_{j,s_j(x)} f_{j,s_j(x)}, \tag{4.2}$$

with $0 \leqslant \pi_{j,s_j(x)}$ and $\sum_{j=1}^{k} \pi_{j,s_j(x)} = 1$. The dictionary densities are chosen as multivariate normal, $f_{j,s_j} = \mathcal{N}_p(\mu_{j,s_j}, \Psi_{j,s_j})$, with a factor analytic form chosen for the covariance to reduce dimensionality

$$\Psi_{j,s_j} = \Theta_{j,s_j}\Theta_{j,s_j}^T + \Sigma_{j,s_j} \tag{4.3}$$

with Θ_{j,s_j} being a $p \times \ell_{j,s_j}$ matrix and $\Sigma_{j,s_j} = diag\left(\sigma_{j,s_j}^1, \ldots, \sigma_{j,s_j}^p\right)$. The covariance decomposition in 4.3 can be induced through the latent factor model

$$y_i = \mu_{j,s_j} + \Theta_{j,s_j}\eta_i + \epsilon_i, \quad \eta_i \sim \mathcal{N}_{\ell_{j,s_j}}(0, I), \quad \epsilon_i \sim \mathcal{N}_p(0, \Sigma_{j,s_j}). \tag{4.4}$$

Therefore, for each node (j, s_j), f_{j,s_j} will be induced marginalizing out the latent variable η_i in the factor model specific to that node. The number of columns of Θ_{j,s_j} (number of factors) varies across nodes and is estimated through an adaptive Gibbs sampler (see §4.2.2).

For the probability weights on the dictionary densities corresponding to each path through the tree, we choose the novel stick-breaking process defined in chapter 3. For each node (j, s_j) in the partition tree, define a stick length $V_{j,s_j} \sim Beta(1, \alpha)$ for nodes (j, s_j) located from generation 1 to $k-1$. The parameter α encodes the complexity of the model, with $\alpha = 0$ corresponding to the case in which $f(y|x) = f(y)$. We relate the weights in (4.2) to the stick-breaking random variables as follows:

$$\pi_{j,s_j} = V_{j,s_j} \prod_{\zeta \in \mathcal{A}(j,s_j)} [1 - V_\zeta],$$

with $V_{k,s_j} = 1$ for any $s_j \in \{1, \ldots, \mathcal{K}_j\}$. This condition will ensure that $\sum_{j=1}^{k} \pi_{j,s_j} = 1$ for any path $\{s_1, \ldots, s_k\}$.

4.2 Estimation

We first partition observations in a tree fashion applying recursively METIS (Karypis and Kumar, 1999) until the stopping criteria presented in chapter 3 is satisfied (refer to section §3.2 for further details). Then, the sequence of dictionary densities $\{f_{j,s_j}\}$ and stick-breaking weights $\{V_{j,s_j}\}$ are estimated using Bayesian methods and inference is carried out using the Gibbs sampler. Under model 4.4, estimating the sequence of dictionary densities is equivalent to estimate sequences $\left\{\Lambda_{j,s_j}\right\}$, $\left\{\mu_{j,s_j}\right\}$ and $\left\{\Sigma_{j,s_j}\right\}$.

4.2.1 Prior Specification

Following Bhattacharya and Dunson (2011), we will consider a shrinkage priors for the columns of the factor loadings in 4.4. Define $\theta_{j,s_j}^{h\iota}$ as the (h,ι) element of Θ_{j,s_j} and consider the following prior specification

$$\theta_{j,s_j}^{(h\iota)} \sim \mathcal{N}\left(0, \frac{1}{\tau_{j,s_j}^{(\iota)}\rho_{j,s_j}^{(h\iota)}}\right), \quad \tau_{j,s_j}^{(\iota)} = \prod_{d=1}^{\iota}\phi_{j,s_j}^{(d)}$$

$$\rho_{j,s_j}^{(h\iota)} \sim Ga(a_1,a_2), \quad \phi_{j,s_j}^{(1)} \sim Ga(a_3,1), \quad \phi_{j,s_j}^{(h)} \sim Ga(a_4,1) \quad \forall h>1$$

Choosing $a_4 > 1$ implies that $\tau_{j,s_j}^{(\iota)}$ stochastically increases with ι, shrinking the elements of Θ_{j,s_j} toward zero increasingly as the number of columns grows. This prior specification allows an adaptive choice of the number of factors. We assign a standard normal density for the intercept parameter μ_{j,s_j} associated to each node (j,s_j), i.e. $\mu_{j,s_j} \sim \mathcal{N}_p(0,I)$. Finally, we specify the prior for Σ_{j,s_j} via the usual inverse gamma priors on its diagonal elements, i.e. $\sigma_{j,s_j}^{(h)} \sim \mathcal{IG}(\alpha_\sigma,\beta_\sigma)$ for $h \in \{1,\ldots,p\}$.

4.2.2 Selection of the Number of Factors

In order to select the number of factors, we directly apply the method proposed by Bhattacharya and Dunson (2011). Basically, the number of factors will be adapted as the Gibbs sampler progresses, with adaptation designed to satisfy the diminishing adaptation condition in Theorem 5 of Roberts and Rosenthal (2007). Directly following Bhattacharya and Dunson (2011), at the hth iteration, adaptation occurs with probability $p(h) = \exp(-(t_1 + t_2 h))$ and (t_1, t_2) chosen such that adaptation occurs every ten iterations at the beginning of the chain and decreases exponentially in the number of iterations. When adaption occurs, the number of columns having all elements in some a priori chosen neighborhood of zero, i.e. $(-\tilde{\epsilon}, +\tilde{\epsilon})$ with $\tilde{\epsilon}$ close to zero, are counted. We can intuitively assume that the factors corresponding to such columns have a negligible contribution, therefore we discard these columns and continue the sampler with a reduced number of factors. Otherwise, if the number of such columns drops to zero we may be missing important factors, therefore we add a column to the loading. The other parameters are modified accordingly and, when a factor is added, the new parameters are sampled from the prior. This adaptation scheme was thought for a single factor model but it can be easily implemented for a mixture of factor analyzers and consequently for our model. When adaptation occurs columns of all loading matrices Θ_{j,s_j}s will be monitored and the number of factors of each loading will be either decreased or increased.

4.2.3 Full Conditionals and Gibbs Sampler Steps

For ease of explanation, we assume in this section a fixed number of factors. The number of levels k and, consequently the number of subsets in $\mathcal{T}$, strictly depends on the number of observations. Therefore, especially for large sample sizes, the dimensionality of the parameter space may become huge and it may lead to computational problems. To solve this computational issue, we implement the slice sampling al-

gorithms proposed by Kalli et al. (2011). For each observation a latent variable $u \in [0,1]$ is introduced and the vector $(y^T, u)^T$ is modeled as

$$f(y,u|x) = \sum_{j=1}^{k} 1\left(u < \pi_{j,s_j(x)}\right) f_{j,s_j(x)}, \tag{4.5}$$

with $1(A)$ being equal to one if the event A occurs. Notice that marginalizing out the latent variable u, the mixture density in 4.5 is recovered. Under model 4.5, components with weights close to zero, and therefore considered unnecessary, will be automatically excluded. Let $g_i \in \{1,\ldots,k\}$ be the indicator variable indicating the level used by observation y_i. Define $\mathcal{I}_{j,m} = \{i : g_i = j,\ x_i \in \mathcal{C}_{j,m}\}$ with $n_{j,m}$ being the cardinality of $\mathcal{I}_{j,m}$. Considering model 4.5 and the prior specification in §4.2.1, the Gibbs sampler iterates through the following steps.

Step 1. Sample u_i for $i \in \{1,\ldots,n\}$

$$\Pr(u_i \,|\, -) = Un\left(0, p_{g_i,s_{g_i}}\right)$$

Step 2. For each node (j,s), with $j = \{1,\ldots,k\}$ and $s \in \{1,\ldots,\mathcal{K}_j\}$, denote $\theta_{j,s}^{(h)}$ the jth row of $\Theta_{j,s}$ and sample $\theta_{j,s}^{(h)}$, for $h \in \{1,\ldots,p\}$ from:

$$\Pr(\theta_{j,s}^{(h)} \,|\, -) = \mathcal{N}_{\ell_{j,s}}\left(\bar{D}_h^{-1} \sum_{i \in \mathcal{I}_{j,s}} \eta_i \left(y_i^{(h)} - \mu_{j,s}^{(h)}\right)/\sigma_j^{(h)}, s, \bar{D}_h^{-1}\right)$$

with $\mu_{j,s}^{(h)}$ being the hth element of $\mu_{j,s}$, $y_i^{(h)}$ being the hth element of y_i, $\bar{D}_h = \left(D_h + A_i/\sigma_j^{(h)}, s\right)$ with $D_h = diag\left(\rho_{j,s}^{(h1)}\tau_{j,s}^{(1)},\ldots,\rho_{j,s}^{(hk)}\tau_{j,s}^{(h)}\right)$ and $A_i = \sum_{i \in \mathcal{I}_{j,s}} \eta_i \eta_i^T$.

Step 3. Update η_i, for $i \in \{1,\ldots,n\}$, from:

$$\Pr(\eta_i \mid -) = \mathcal{N}_{\ell_{g_i,s_{g_i}}} \left(\bar{D}_\eta^{-1} \Theta_{g_i,s_{g_i}}^T \Sigma_{g_i,s_{g_i}}^{-1} (y_i - \mu_{g_i,s_{g_i}}), \bar{D}_\eta^{-1} \right)$$

with $\bar{D}_\eta = \left(I + \Theta_{g_i,s_{g_i}}^T \Sigma_{g_i,s_{g_i}}^{-1} \Theta_{g_i,s_{g_i}} \right)$.

Step 4. Update $\sigma_{j,s}^{(h)}$, for $h \in \{1, \ldots, p\}$ and $j = \{1, \ldots, k\}$ and $s \in \{1, \ldots, \mathcal{K}_j\}$, from:

$$\Pr\left(\sigma_{j,s}^{(h)} \mid -\right) = \mathcal{IG}\left(a_\sigma + n_{j,s}/2, b_\sigma + \sum_{i \in \mathcal{I}_{j,s}} \left(y_{ih} - \mu_{j,s}^{(h)} - \theta_{j,s}^{(h)} \eta_i \right)^2 \right)$$

Step 5. Update $\mu_{j,s}$, for $j = \{1, \ldots, k\}$ and $s \in \{1, \ldots, \mathcal{K}_j\}$, from

$$\Pr\left(\mu_{j,s} \mid -\right) = \mathcal{N}_p \left(\left(n_{j,s} \Sigma_{j,s}^{-1} + I \right)^{-1} \sum_{i \in \mathcal{I}_{j,s}} \left(y_i - \Theta_{j,s} \eta_i \right), \left(n_{j,s} \Sigma_{j,s}^{-1} + I \right)^{-1} \right)$$

Step 6. Update g_i by sampling from the multinomial full conditional with

$$\Pr(g_i = j \mid -) \propto 1\left(u_i < p_{j,s_j(x_i)} \right) f_{j,s_j(x_i)}(y_i)$$

with $f_{j,s_j(x_i)} = \mathcal{N}\left(\mu_{j,s_j(x_i)}, \Theta_{j,s_j(x_i)} \Theta_{j,s_j(x_i)}^T + \Theta_{j,s_j(x_i)} \right)$

Step 7. Update stick-breaking random variable $V_{j,s}$ from

$$\Pr(V_{j,s} \mid -) = \text{Beta}\left(1 + n_{j,s}, \alpha + \sum_{v \in \mathcal{D}(j,s)} n_v \right)$$

for any set j, s located from tree level 1 to $k-1$. Let $V_{k,s} = 1$ for all s and let $\pi_{j,s} = V_{j,s} \prod_{v \in \mathcal{A}(j,s)} (1 - V_v)$. In particular, $\mathcal{A}(j,s)$ and $\mathcal{D}(j,s)$ are defined as the set of ancestors and descendants of node (j,s) as in chapter 3.

Step 8. Update $\rho_{j,s}^{(h,\iota)}$, for $h \leqslant p$, $\iota \leqslant \ell_{j,s}$, $j \leqslant k$ and $s \in \{1, \ldots, \mathcal{K}_j\}$ from

$$\Pr(\rho_{j,s}^{(h,\iota)} \mid -) = \mathcal{G}\left(a_1 + 1/2, a_2 + \tau_{j,s}^i \left(\theta_{j,s}^{(h,\iota)} \right)^2 / 2 \right)$$

Step 9. Update $\phi^1_{j,s}$, $j \leqslant k$ and $s \in \{1, \ldots, \mathcal{K}_j\}$ from

$$\Pr(\phi^{(1)}_{j,s} \mid -) = \mathcal{G}\left(a_3 + p\ell_{j,s}/2, 1 + \sum_{h=1}^{\ell_{j,s}} \tau^{(h)}_{j,s} \sum_{w=1}^{p} \rho^{(wh)}_{(js)} \left(\theta^{(wh)}_{j,s}\right)^2/2\right)$$

Step 10. Update $\phi^d_{j,s}$, for $1 < d \leqslant \ell_{j,s}$, $j \leqslant k$ and $s \in \{1, \ldots, \mathcal{K}_j\}$ from

$$\Pr(\phi^{(d)}_{j,s} \mid -) = \mathcal{G}\left(a_4 + (\ell_{j,s} - d + 1)p/2, 1 + \sum_{h=d}^{\ell_{j,s}} \tau^{(h)}_{j,s} \sum_{w=1}^{p} \rho^{(w,h)}_{(js)} \left(\theta^{(wh)}_{j,s}\right)^2/2\right)$$

In order to proceed with the chain, it is not required to sample all $(\mu_{j,s}, \Lambda_{j,s}, \Sigma_{j,s})$s. We only need to sample parameters necessary to do *Step 6* exactly. Therefore, only parameters involved in components with non-negligible weights will need to be sampled. This will reduce dramatically the computational burden of the proposed algorithm.

4.3 Synthetic Example

In the following simulation examples, we test the predictive performance of the proposed model relative to competing alternatives. We initially consider the case in which x_i is a two dimensional vector defined over a bounded set and then move to the more general case in which $x_i \in \Re^q$. In all examples, the proposed model will be compared to mixture of factor analyzers (MFA) and covariate dependent mixture of factor analyzers (dMFA). The latter provides additional flexibility by allowing the mixing weights to change flexibly with covariates. In particular, covariate dependent weights will be modeled through the probit stick breaking process (Rodriguez and Dunson, 2011). The three models will be compared in terms of mean squared error in leave-one-out cross validation. Predictions will be carried out using the same methodology described in §3.2.2. We used $10,000$ Gibbs sampling iterations and a burn-in of $1,000$. We set the hyperparameters at the following fixed non-optimized

values for all simulations and real data experiments and do not tune: $a_1 = a_2 = 2$, $a_3 = 1$, $a_4 = 4$, $\alpha_\sigma = 1$, $\beta_\sigma = 0.3$ and $\alpha = 1$. We set parameters used to select the number of factors as $\tilde{\epsilon} = 0.01$, $t_1 = 1$ and $t_2 = 5 \times 10^{-4}$.

4.3.1 Two Dimensional Predictors

We initially assume x_i is sampled iid from a uniform distribution over$[0,1]^2$. In the first simulation scenario, y_i is drawn from a feature-dependent Gaussian density. Specifically,

$$y_i \sim \mathcal{N}_p(\mu_0(x_i), \Psi_0(x_i)), \qquad \Psi_0(x_i) = \Lambda_0(x_i)\Lambda_0(x_i)^T + \Sigma_0$$

with Λ_0 being a $p \times \ell_0$ matrix, Σ_0 a $p \times p$ identity matrix and $\ell_0 << p$. We generated $\mu_0(x_i)$ and $\Lambda_0(x_i)$ as follows

$$(\mu_{0h}(x_1), \ldots, \mu_{0h}(x_n)) \overset{iid}{\sim} \mathcal{N}_n\left(0, C(x)\right), \quad h \in \{1, \ldots, p\}$$

$$\left(\lambda^0_{js}(x_1), \ldots, \lambda^0_{js}(x_n)\right) \overset{iid}{\sim} \mathcal{N}_n\left(0, C(x)\right), \quad j \in \{1, \ldots, p\}, s \in \{1, \ldots, \ell_0\}$$

with $\mu_{0h}(x_i)$ and $\lambda^0_{js}(x_i)$ being respectively the hth element of $\mu_0(x_i)$ and the (j,s)th element of $\Lambda_0(x_i)$. We define the (j,h)th element of $C(x)$ as $C_{jh}(x) = \upsilon \exp\{-d(x_j, x_h)\}$ with $d(x_j, x_h)$ being the Euclidean distance between vectors x_j and x_h, $\upsilon = 2$, and the number of factors equal to $\ell_0 = 5$ or $\ell_0 = 10$.

In the second simulation scenario, y_i is drawn from a mixture of factor analyzers with feature-dependent mixture weights. Let $k_0 \in \{3, 5\}$ be the number of mixture components and let $\Phi(\cdot)$ be the cumulative distribution function of the standard normal density. Mixture weights were derived through a probit stick breaking specification (Rodriguez and Dunson, 2011). Model stick breaking weights as

$$\pi_{0j}(x_i) = V_{0j}(x_i) \prod_{s=1}^{j-1} \left(1 - V_{0s}(x_i)\right)$$

with $V_{0j}(x_i) = \Phi(\vartheta_{j1}x_{i1} + \vartheta_{j2}x_{i2})$ for $j < k_0$ and $V_{0k_0} = 1$. Let μ_{0j} and $\Psi_{0j} = \Lambda_{0j}\Lambda_{0j}' + \Sigma_{0j}$ be respectively the intercept and the covariance of the jth mixture component, with Λ_{0j} being a $p \times \ell_0$ loading matrix and Σ_{0j} a $p \times p$ diagonal matrix with positive entries on the diagonal. In particular, sample each element of the loading Λ_{0j} and μ_{0j}, for $j \in \{1, \ldots, k_0\}$, independently from a normal with mean m_{0j} and unitary variance. For $k_0 = 3$, we set

$$m_0 = (2, 0, -2) \quad , \quad (\vartheta_{11}, \vartheta_{21}) = (0.5, 0) \quad , \quad (\vartheta_{12}, \vartheta_{22}) = (-0.5, 0)$$

while for $k_0 = 5$ we set

$$m_0 = (2, 0, -2, -1, 1) \quad , \quad (\vartheta_{11}, \vartheta_{21}, \vartheta_{31}, \vartheta_{41}) = (0.5, 0, -1, 1)$$

$$(\vartheta_{12}, \vartheta_{22}, \vartheta_{32}, \vartheta_{42}) = (-0.5, 0, 1, -1)$$

The diagonal elements of Σ_{0j} are drawn from an inverse Gamma density with scale and rate parameter equal to 5 and 3 respectively. The number of factors was set equal to 5.

Table 4.1 shows mean squared errors based on leave-one-out predictions. In essentially every case considered, our Bayesian factor trees (BFT) approach produced the lowest MSE followed by dMFA, with MFA having the worst performance. In the second simulation scenario, when observations are sampled from a dMFA, our approach and dMFA perform similarly.

To visualize the performance of our model, we define a grid of 100 evenly spaced points in $\mathcal{U} = [0, 1]^2$. We sampled $y_i \in \Re^p$ with $p = 50$ and $i \in \mathcal{U}$ from the above three component mixture of factor analyzers. Figure 4.1 shows the true value and the estimate of five variables in y_i for each $i \in \mathcal{U}$. At each Markov Chain iteration, we sampled the five variables conditionally on the other $p-5$ variables, then in figure 4.1 we plot the mean of those values over Markov chain iterations. As shown, the proposed model performs similarly to dMFA in estimating elements of y_i, while MFA

is not able to capture most of the spatial structure. The associated mean squared errors were $(\hat{\epsilon}^2_{MFA}, \hat{\epsilon}^2_{dMFA}, \hat{\epsilon}^2_{BFT}) = (0.33, 0.27, 0.27)$, revealing that we were able to perform similarly to dMFA.

Table 4.1: Two dimensional predictors: Mean and standard deviations of squared errors under our bayesian factor tree (BFT), a mixture of factor analyzers (MFA) and covariate dependent mixture of factor analyzers (dMFA) under the first (1) and second (2) simulation scenario. **Bold** indicates best MSE. As shown, in almost all data scenarios, BFT leads to the lowest MSE.

				$k_0 = 3$			$k_0 = 5$	
SIM	p	n	BFT	MFA	dMFA	BFT	MFA	dMFA
(1)	100	50	**0.11**	0.15	**0.11**	0.14	0.19	0.20
			(0.09)	(0.12)	**(0.07)**	(0.08)	(0.17)	(0.15)
		100	**0.06**	0.33	0.13	**0.08**	0.17	0.13
			(0.02)	(0.46)	(0.06)	**(0.03)**	(0.14)	(0.07)
	500	50	**0.08**	0.16	0.11	**0.08**	0.21	0.20
			(0.06)	(0.12)	(0.13)	**(0.06)**	(0.16)	(0.11)
		100	**0.05**	0.12	0.10	**0.14**	0.20	0.18
			(0.05)	(0.11)	(0.09)	**(0.14)**	(0.16)	(0.16)
				$\ell_0 = 5$			$\ell_0 = 10$	
SIM	p	n	BFT	MFA	dMFA	BFT	MFA	dMFA
(2)	100	100	**0.14**	0.25	0.25	**0.26**	0.27	0.29
			(0.09)	(0.06)	(0.09)	**(0.14)**	(0.18)	(0.28)
		200	**0.16**	0.20	0.19	**0.17**	0.19	0.19
			(0.11)	(0.16)	(0.16)	**(0.08)**	(0.10)	(0.13)
	500	100	**0.14**	0.16	0.15	**0.22**	0.23	0.26
			(0.06)	(0.05)	(0.05)	**(0.12)**	(0.14)	(0.17)
		200	**0.11**	0.15	0.12	0.25	0.26	**0.19**
			(0.06)	(0.07)	(0.04)	(0.21)	(0.24)	**(0.10)**

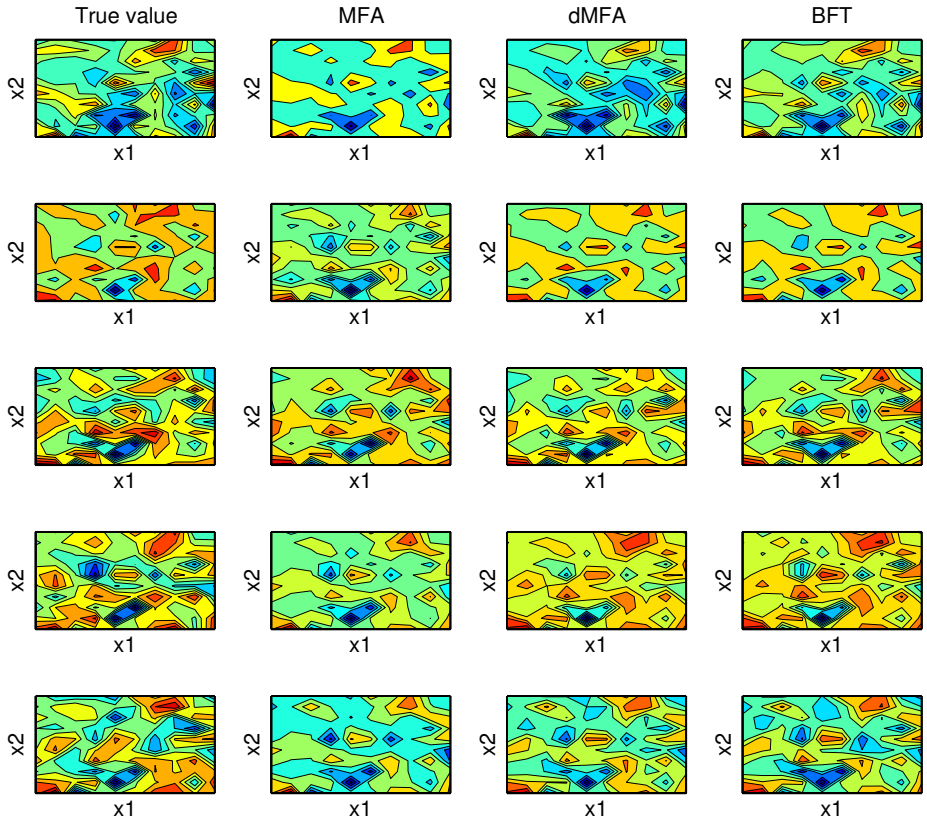

FIGURE 4.1: True value and estimate under MFA, dMFA and BFT of five variables of y_i given $x_i = (x_{i1}, x_{i2})^T$ for $i = \{1, \ldots, 100\}$. Each row correspond to a different element of the response vector y, while each column correspond to a different method utilized to predict y. As shown, BFT performs similarly to dMFA in estimating the five elements of y, while a simple MFA (not depending on covariates) is not able to capture most of the spatial structure.

4.3.2 Higher Dimensional Predictors

In this section we consider examples involving a large number of predictors. In all scenarios, the response variable y_i was sampled from a mixture of factor analyzers with feature-dependent weights (see §4.3.1). Two different models are considered for the feature vector. First, we assume $x_i \sim \mathcal{N}_q(0, \Psi)$, with (s, j)-element of Ψ defined as $\Psi_{sj} = j, s\varrho^{|s-j|}$, $\varrho = 0.9$ and $j, s = 1$. Given x_i, y_i is sampled from a mixture of

factor analyzers with stick breaking weights depending only on the first covariate x_{i1}, i.e. $V_{0j}(x_i) = \Phi(\vartheta_j x_{i1})$ for $j < k_0$ and $V_{0k_0}(x_i) = 1$. In the second data scenario, x_i is sampled from the Swissroll manifold, a two dimensional manifold embedded in $\Re^q$ (see figure 3.3(a)). In this case, stick breaking weights are assumed to depend only on one coordinate of the Swissroll, i.e. $V_{0j}(x_i) = \Phi(\vartheta_j s_{i1})$ for $j < k_0$ and $V_{0k_0}(x_i) = 1$ with s_{i1} being a coordinate of the manifold. For both scenarios, we consider $k_0 = 3$ and $(\vartheta_1, \vartheta_2) = (-2, 2)$. The intercepts and loading matrices are drawn considering the same model as in §4.3.1.

We sampled 20 datasets for each data scenario. Table 4.2 shows mean squared errors based on leave-one-out predictions under experiments involving different combination of (n, p, q). As shown, in almost all data scenarios, our model is able to perform as well as or better than the model associated to the lowest mean squared error. As expected, the dependent MFA leads to better MSE compared to MFA. Figure 4.2 shows the relative CPU time of our approach versus dMFA. The relative CPU time was computed as in chapter 3, i.e. $r_m^{\mathcal{A}} = \phi(MSB)/\phi(\mathcal{A})$, where ϕ is the CPU time in seconds and $\mathcal{A}$ is the competitor algorithm. As shown, our approach can scale substantially better than dMFA to large number of predictors. Notice, that in this section we have considered examples involving few thousands of predictors. However, in many real world applications the number of predictors can grow up to hundreds of thousands. In this framework, a dependent MFA becomes computationally prohibitive.

4.4 Real Application

In order to test the predictive performance of the proposed model, we considered two datasets involving a moderately high number of features. The first real data experiment comprises 40 genes (response) involved in the isoprenoid pathway of Arabidopsis thaliana. This set of genes were found to be highly correlated with 795

Table 4.2: Higher dimensional predictors: Mean and standard deviations of squared errors under our bayesian factor tree (BFT), a mixture of factor analyzers (MFA) and covariate dependent mixture of factor analyzers (dMFA). **Bold** indicates best MSE.

			Normal model			Swissroll		
p	q	n	BFT	MFA	dMFA	BFT	MFA	dMFA
100	1,000	50	**0.10**	0.31	0.29	**0.09**	0.28	0.13
			(0.09)	(0.29)	(0.27)	**(0.06)**	(0.32)	(0.19)
		100	**0.18**	0.49	0.29	**0.15**	0.41	0.20
			(0.20)	(0.45)	(0.30)	**(0.28)**	(0.73)	(0.35)
	5,000	50	**0.14**	0.26	0.20	**0.21**	0.37	0.29
			(0.10)	(0.24)	(0.84)	**(0.25)**	(0.63)	(0.45)
		100	**0.18**	0.40	0.33	**0.14**	0.44	0.20
			(0.29)	(0.57)	(0.21)	**(0.09)**	(0.66)	(0.61)
	10,000	50	**0.08**	0.56	0.44	**0.25**	0.47	0.36
			(0.03)	(0.98)	(0.56)	**(0.23)**	(0.20)	(0.24)
		100	**0.30**	0.44	0.35	**0.26**	0.59	0.42
			(0.22)	(0.32)	(0.32)	**(0.14)**	(0.45)	(0.30)
500	1,000	50	**0.45**	0.57	0.47	**0.43**	0.93	0.76
			(0.17)	(0.31)	(0.68)	**(0.22)**	(0.64)	(0.96)
		100	**0.40**	0.83	0.79	**0.44**	0.67	0.50
			(0.28)	(0.61)	(0.47)	**(0.23)**	(0.48)	(0.90)
	5,000	50	0.62	**0.59**	0.58	**0.92**	0.96	0.94
			(0.63)	**(0.37)**	(0.33)	**(0.34)**	(0.31)	(0.34)
		100	**0.35**	0.59	0.59	**0.85**	0.88	0.73
			(0.15)	(0.23)	(0.66)	**(0.28)**	(0.28)	(0.33)
	10,000	50	**0.33**	0.58	0.42	**0.29**	0.83	0.70
			(0.32)	(0.34)	(0.35)	**(0.16)**	(1.08)	(0.68)
		100	**0.33**	0.36	0.35	0.40	0.34	**0.32**
			(0.27)	(0.25)	(0.26)	(0.30)	(0.23)	**(0.21)**

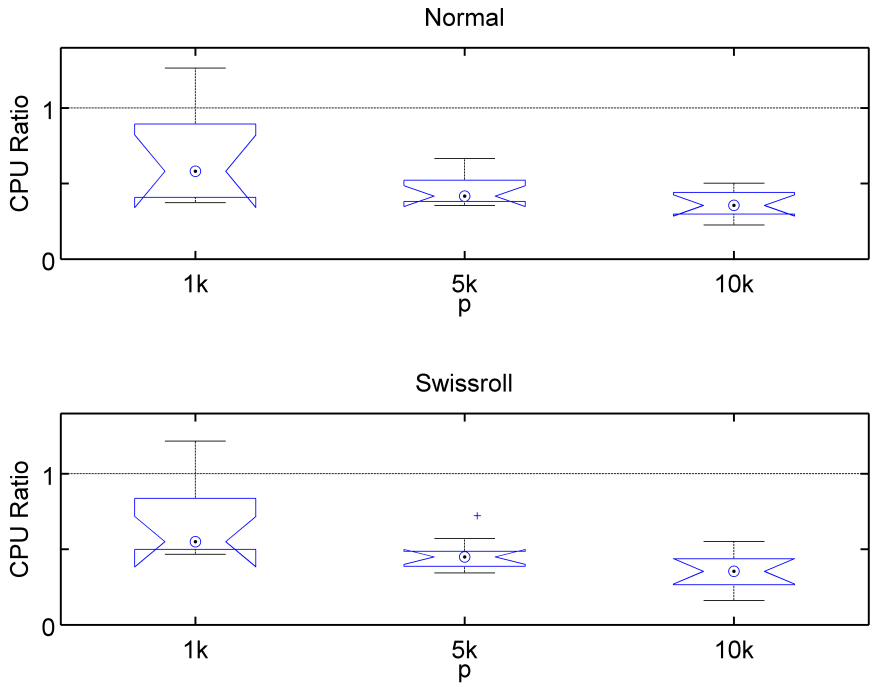

FIGURE 4.2: Plots depicts the relative CPU time of BFT (our approach), versus dMFA as a function of ambient dimension of x, under the normal and the swissroll simulation scenario with $q = 500$ and $n = 100$. The x-axis is the number of predictors involved in the experiment, where k equals 1 thousand, so that 2k=2,000. BFT outperforms dMFA regardless of ambient dimension ($r_{cpu} < 1$ for all p).

genes (predictors) from 56 other metabolic pathways in Arabidopsis thaliana (Wille et al., 2004). All variables have been log-transformed and standardized to zero mean and unit variance.

The second dataset is a large population dataset involving 518 subjects from the capital region of Finland. For each subject a set of 138 metabolites and about $35,000$ genes are measured. Inouye et al. (2010) identified a set of highly correlated genes, the lipidleukocyte (LL) module, as having a prominent role in over 80 metabolites. Therefore, gene levels should be informative about the two third of metabolites. All variables were standardized to zero mean and unit variance.

The predictive accuracy of each method was estimated by leave-one-out cross-

validation. Table 4.3 shows percentiles of squared errors for the two data examples above. For the first data example, we compare our model to MFA and dMFA. For the second data example, given the ultra-high dimensionality of the predictors space, we only compared our approach to MFA. As shown, our approach leads to the best predictive performance.

Table 4.3: Real dataset: Percentiles (2.5%, 50% and 97.5%) of squared errors under our Bayesian Factor Tree (BFT), a mixture of factor analyzers (MFA) and covariate dependent mixture of factor analyzers (dMFA). For the second data example, given the ultra-high dimensionality of the predictor space, we compared our approach only to MFA.

	n	p	q	MFA (2.5%, 50%, 97.5%)	BFT (2.5%, 50%, 97.5%)	dMFA (2.5%, 50%, 97.5%)
(1)	118	40	795	(0.50, 0.97, 1.15)	(0.40, 0.74, 0.90)	(0.45, 0.89, 0.95)
(2)	518	138	35K	(0.60, 0.87, 1.20)	(0.50, 0.71, 0.95)	

5

Concluding Remarks and Future Direction

To summarize, we have dealt with two problems in this thesis. For the first problem we have proposed a new repulsive mixture modeling framework, which should lead to substantially improved unsupervised learning (clustering) and density estimation performance in general applications. A key aspect is soft penalization of components located close together to favor, without sharply enforcing, well separated clusters that should be more likely to correspond to the true missing labels. We have focused on Bayesian MCMC-based methods, but there are numerous interesting directions for ongoing research, including fast optimization-based approaches for learning mixture models with repulsive penalties.

The other problem, we have dealt with, is to learn a the density of a response variable given high dimensional features. In chapter 3, we have introduced a general formalism to estimate conditional distributions via multiscale dictionary learning. We developed a novel multiresolution stick breaking process that can scale substantially better than other existing algorithms to massive number of features, while resulting in good predictive performance. An important property of any such strategy is the ability to scale up to ultrahigh-dimensional predictors. We considered

simulations and real-data examples where the dimensionality of the predictor space exceeded several thousands. To our knowledge, no other approach to learn conditional distributions can run at this scale. Our approach explicitly assumes that the posterior $f(y|x)$ can be well approximated by projecting x onto a lower-dimensional space. Note that this assumption is much less restrictive than assuming that x is close to a low-dimensional space; rather, we only assume that the part of $f(x)$ that "matters" to predict y lives near a low-dimensional subspace. Because a fully Bayesian strategy remains computationally intractable at this scale, we developed an empirical Bayes approach, estimating the partition tree based on the data, but integrating over scales and posteriors.

We demonstrate that even though we obtain posteriors over the conditional distribution $f(y|x)$, our approach, dubbed multiscale stick-breaking (MSB), outperforms standard machine learning algorithms in terms of both predictive accuracy and computational time, as the sample size and ambient dimension increase. In future work, we will extend these numerical results to obtain theory on posterior convergence. Indeed, while multiscale methods benefit from a rich theoretical foundation (Allard et al., 2012), the relative advantages and disadvantages of a fully Bayesian approach, in which one can estimate posteriors over all functionals of $f(y|x)$ at all scales, remains relatively unexplored.

In chapter 4, we have extended the multiresolution stick breaking model proposed in chapter 3 to handle multivariate responses. For this purpose, dictionary densities were defined as multivariate normal with a factor analytic form chosen for the covariance to reduce dimensionality. The proposed model results in a mixture of factor analyzers defined over different levels of resolution. As illustrated, inference on component-specific parameters is carried out using Gibbs sampler. Our model leads to good predictive performance and can scale to high number of features. However, the proposed model may face computational problems as the number of response

variables increases. In fact, at each Gibbs sampler step and for each observation, the likelihood function (a mixture of multivariate Gaussians) needs to be computed. This step can become computational prohibitive as the number of response variables increase, reducing dramatically the efficiency of our model. There are a variety of real worlds applications involving a large number of response variables depending on huge number of features. For this applications, a more efficient algorithm relying on some likelihood approximation needs to be considered. In future works, we will extend our algorithm to efficiently handle situations in which not only the predictors but also the response is high dimensional.

Another possible direction for future work is using parallelized and distributed systems to estimate the proposed multiresolution stick breaking model. Though this model can scale substantially better than competitors to high dimensional features, we may gain more efficiency by using parallelized and distributed systems. For this purpose, we should adopt other estimation techniques rather than Bayesian method relying on Markov chain Monte Carlo. In fact, given the serial structure of MCMC algorithms, they cannot fully be learned using parallelized systems. Alternatively, we may use an hybrid model where dictionary densities are estimated in parallel using frequentist methodologies, such as maximum likelihood estimation, and then stick breaking weights are estimated through Markov chain Monte Carlo.

Appendix A

Chapter 2: Theory

A.1 Cited Theorems and Assumptions

Assumptions of theorem 3.1 in Scricciolo (2011)

(i) The prior on σ has a continuous and positive Lebesgue density ψ on an interval containing σ_0 and its distribution function Ψ, for constants $e_1, e_2, e_3 > 0$, satisfies

$$\Psi(s) \leqslant \exp(-e_1 s^{-e_2}) \text{ as } s \to 0 \text{ and } 1 - \Psi(s) \leqslant s^{-e_3} \text{ as } s \to \infty$$

(ii) The prior for the number of components is such that, for constants $d_1, d_2 > 0$,

$$0 < \vartheta(k) \leqslant d_1 \exp(-d_2 k) \text{ for all } k \in \mathbb{N}$$

(iii) For each k, the prior for the weights is a Dirichlet with parameters $(\alpha_1, \ldots, \alpha_k)$ such that, for constants a_1, $a_2 > 0$, $a_3 \geqslant 1$ and for $0 < \epsilon \leqslant 1/(a_3 k)$ and $j = 1, \ldots, k$

$$a_2 \epsilon^{a_1} \leqslant \alpha_j \leqslant a_3$$

Assumptions B1-B5

Assumptions B1-B5 corresponds to assumptions A1-A5 in Rousseau and Mengersen

(2011). Assumptions differ only in the conditions concerning the prior on the component-specific parameters in assumption A5. In condition B5, we assume that π is defined as (2) and h is defined as either (3) or (4). For the sake of clarity, let us state assumption B1:

B1) There exists a $q \geqslant 0$ such that for $\delta_n = (\log n)^q n^{-1/2}$ the following holds

$$\lim_{M\to\infty} \limsup_{n\to\infty} E_n^0 \left\{\Pi\left(\|f - f_0\|_1 \geqslant M\delta_n | Y_n\right)\right\} = 0$$

Ghosal et al. (2000)'s Theorem

Theorem 8. *Let π_n be a sequence of priors on a class of densities $\mathcal{F}$ equipped with a metric d that can be either the Hellinger or the one induced by the L_1-norm. Assume that for positive sequences $\bar{\epsilon}_n, \tilde{\epsilon}_n \to 0$ such that $n\min(\bar{\epsilon}_n, \tilde{\epsilon}_n) \to \infty$, constants $d_1, d_2, d_3, d_4 > 0$ and sets $\mathcal{F}_n \subseteq \mathcal{F}$, we have*

$$\log D(\bar{\epsilon}_n, \mathcal{F}_n, d) \leqslant d_1 n\bar{\epsilon}_n^2 \tag{A.1}$$

$$\pi_n(\mathcal{F}\backslash\mathcal{F}_n) \leqslant d_3 \exp\left\{-(d_2+4)n\tilde{\epsilon}_n^2\right\} \tag{A.2}$$

$$\pi_n\left\{B_{K-L}(f_0; \tilde{\epsilon}_n^2)\right\} \geqslant d_4 \exp(-d_2 n\tilde{\epsilon}_n^2) \tag{A.3}$$

where $B_{K-L}(f_0; \tilde{\epsilon}_n^2) = \left\{f : \int f_0 \log(f_0/f) \leqslant \tilde{\epsilon}_n^2; \int f_0 \log(f_0/f)^2 \leqslant \tilde{\epsilon}_n^2\right\}$.
Then, for $\epsilon_n = \max(\bar{\epsilon}_n, \tilde{\epsilon}_n)$ and a sufficiently large constant $M > 0$, the posterior probability

$$\pi_n\{f : d(f, f_0) > M\epsilon_n | Y_n\} \to 0$$

in P_0^n probability, as $n \to \infty$.

A.2 Proofs

Throughout the appendix we write all constants whose values are of no consequence to be equal to 1.

Proof of lemma 1. By assumption B0, $\vartheta(k = k_0) > 0$. We consider the case f is a finite mixture with k_0 components. By assumption A1, for each $\eta > 0$ there is a corresponding $\delta > 0$ such that, for any given $y \in \mathcal{Y}$ and for all $\gamma_1, \gamma_2 \in \Gamma$ with $|\gamma_1 - \gamma_2| < \delta$, we have that $|\phi(y;\gamma_1) - \phi(y;\gamma_2)| < \eta$. Let $S_\delta = P_\delta \times \Gamma_\delta$ with $\Gamma_\delta = \{\gamma : |\gamma_j - \gamma_{0j}| \leqslant \delta, j \leqslant k_0\}$ and $P_\delta = \{p : |p_j - p_{0j}| \leqslant \delta, j \leqslant k_0\}$. By assumption A1 and A2, for any given y and for any $\eta > 0$, there is a $\delta > 0$ such that $|f_0 - f| \leqslant \eta$ if $\theta \in S_\delta$. This means that, $f \to f_0$ as $\theta \to \theta_0$, for any given y. Equivalently, we can say that $|\log(f_0/f)| \to 0$ pointwise as $\theta \to \theta_0$. Notice that

$$|\log(f_0/f)| \leqslant \left|\log\left\{\sup_{\gamma \in D_0} \phi(\gamma)\right\} - \log\left\{\inf_{\gamma \in D_0} \phi(\gamma)\right\}\right|$$

By assumption A3 and applying the dominated convergence theorem, for any $\epsilon > 0$ there is a $\delta > 0$ such that $\int f_0 \log(f_0/f) < \epsilon$ if $\theta \in S_\delta$. By the independence of the weights and the parameters of the kernel,

$$\Pi(KL(f_0, f) < \epsilon) \geqslant \lambda(P_\delta)\pi(\Gamma_\delta)$$

Assumption A4 combined with the fact that $\{\gamma : ||\gamma - \gamma_0||_1 \leqslant \delta\} \subseteq \Gamma_\delta$ result in $\pi(\Gamma_\delta) > 0$. Finally, since $\lambda = Dirichlet(\alpha)$, it can be shown that $\lambda(P_\delta) > 0$. □

Proof of lemma 2. Recall that, under assumptions in lemma 1, γ is a vector of only location parameters. For any given $x \in \Gamma^k$, define $D_x = \{\gamma : ||\gamma - x||_1 < \upsilon/2\}$. By the assumptions on h, for any given x satisfying condition A4 in lemma 2, $h(\gamma) > 0$ for γ such that $d(\gamma_s, x_s) < \upsilon/2$ for $s = 1, \ldots, k$. Since,

$$D \subseteq \{\gamma : d(\gamma_s, x_s) < \upsilon/2; s = 1, \ldots, k\},$$

it follows that $h(\gamma) > 0$ on D. By assumption, g_0 is positive on Γ, therefore it follows that $\pi(\gamma) > 0$ on D. □

Proof of lemma 3. To prove lemma 3 we need to show that the three conditions of theorem 2.1 in Ghosal et al. (2000) are satisfied. First, define $D(\epsilon, \mathcal{F}, d_s)$ as the maximum number of points in $\mathcal{F}$ such that the distance, with respect to metric d_s, between each pair is at least ϵ. Let d_s be either the Hellinger metric or the one induced by the L1-norm. For given sequences $k_n, a_n, u_n \uparrow \infty$ and $b_n \downarrow 0$ define

$$\mathcal{F}_n^{(k)} = \left\{ f : f = \sum_{j=1}^{k} p_j \phi(\gamma_j, \sigma), \gamma \in (-a_n, a_n)^k, \sigma \in (b_n, u_n) \right\}$$

and $\mathcal{F}_n = \cup_{j=1}^{k_n} \mathcal{F}_n^{(j)}$. As it is shown in Scricciolo (2011), for constants $f_2 \geqslant f_1 > 0$ and $l_1, l_2, l_3 > 0$, derived below to satisfy condition (2) and (3) in Ghosal et al. (2000), and defining $f_1 \log n \leqslant k_n \leqslant f_2 \log n$, $a_n = l_3 (\log \bar{\epsilon}_n^{-1})^{1/2}$, $b_n = l_1 (\log \bar{\epsilon}_n^{-1})^{-1/e_2}$ and $u_n = \bar{\epsilon}_n^{-l_2}$, $\log D(\bar{\epsilon}_n, \mathcal{F}_n, d_s) \lesssim n\bar{\epsilon}_n^2$ with $\bar{\epsilon}_n = n^{-1/2} \log n$.

Let $A_{n,j} = (-a_n, a_n)^j$. In order to show condition (2) of theorem 2.1. in Ghosal et al. (2000), we need to show that there is a constant $q_1 > 0$ such that $\pi(A_{n,k}^C) \lesssim \exp(-q_1 a_n^2)$. From the exchangeability assumption it follows

$$\begin{aligned} pr(A_{n,k}^C | k = s) &= \textstyle\sum_{j=1}^{s} \frac{s!}{j!(s-j)!} \pi \left(A_{n,j}^C \times A_{n,s-j} \right) \\ &\leqslant s \textstyle\sum_{j=1}^{s} \frac{(s-1)!}{(j-1)!(s-j)!} \pi \left(A_{n,j}^C \times A_{n,s-j} \right) \leqslant s \pi_m(A_{n,1}^C) \end{aligned}$$

Therefore, condition C1 implies that, for a positive constant q_1 we have $\pi(A_{n,k}^C) \lesssim E(k) \exp(-q_1 a_n^2)$ with $E(k) < \infty$ by condition (ii). Given a positive constant z_2 chosen to satisfy condition (3) in theorem 2.1 of Ghosal et al. (2000), let $f_1 \geqslant (z_2+4)/d_2$, $l_1 \leqslant \{e_1/4(z_2+4)\}^{1/e_2}$, $l_2 \geqslant 4(z_2+4)/e_3$ and $l_3 \geqslant \{4(z_2+4)/q_1\}^{1/2}$. Under these values of f_1, l_1, l_2 and l_3, following Scricciolo (2011), assumptions (i), (ii) and assumption C1 imply $\Pi(\mathcal{F} \backslash \mathcal{F}_n) \lesssim \exp\{-(z_2+4) n \tilde{\epsilon}_n^2\}$ with $\tilde{\epsilon}_n = n^{-1/2} (\log n)^{1/2}$.

To show condition (3) of theorem 2.1 in Ghosal et al. (2000), we can again follow the proof of theorem 3.1. in Scricciolo (2011). The only thing we need to show is

that, there are constants $u_1, u_2, u_3 > 0$ such that for any $\epsilon_n \leqslant u_3$

$$\pi(||\gamma - \gamma_0||_1 \leqslant \epsilon_n) \geqslant u_1 \exp\{-u_2 k_0 \log(1/\epsilon_n)\}$$

that is guaranteed by condition C2. Therefore, it can be easily showed that, for sufficiently large n, $z_2 > 0$ and $\tilde{\epsilon}_n = n^{-1/2}(\log n)^{1/2}$, $\Pi\{B_{KL}(f_0, \tilde{\epsilon}_n^2)\} \gtrsim \exp(-z_2 n \tilde{\epsilon}_n^2)$.

□

Proof of lemma 4. First, let us check that condition C1 is satisfied. Clearly, under the assumptions on h, π leads to exchangeable atoms. Under the assumptions on π, the following holds

$$\pi_m(|\gamma_1| \geqslant t) = \int_{|\gamma_1| \geqslant t} \pi_m(\gamma_1) d\gamma_1 \leqslant c_1 c_2 \int_{|\gamma_1| \geqslant t} g_0(\gamma_1) d\gamma_1$$

with c_1 and c_2 defined as in (2). It follows that there exists a constant $n_1 > 0$ such that $\pi_m(|\gamma_1| \geqslant t) \lesssim \exp(-n_1 t^2)$.

Now let us verify condition C2. Assumptions on h imply that for any $0 < \epsilon < 1$ there is a corresponding $0 < \delta = g^{-1}(\epsilon)$ and constants $w_1 > 0$ such that $h(\gamma) \geqslant w_1 \epsilon^{k_0}$ for all γ satisfying $\min_{\{(s,j):s<j\}} d(\gamma_j, \gamma_s) \geqslant \delta$. Let u_3 be defined as

$$u_3 = \min\left[\epsilon_1/2, g\left(\delta_1\right)\right]$$

with ϵ_1 defined as in assumption B0 and $\delta_1 = \epsilon_1(1 - 1/k_0)$. By assumption $\epsilon < u_3$ and therefore $\delta < \delta_1$. Let us define $M(\gamma, x)$ and $N(\gamma, x)$ as follows,

$$M(\gamma, x) = \left\{\gamma : \min_{\{(s,j):s<j\}} d(\gamma_j, \gamma_s) \geqslant x\right\}, \quad N(\gamma, x) = \{\gamma : |\gamma_j - \gamma_{0j}| \leqslant x; j = 1, \ldots, k_0\}$$

Then,

$$\begin{aligned}\pi(||\gamma-\gamma_0||_1 \leqslant \epsilon) &\geqslant \int_{\{||\gamma-\gamma_0||_1 \leqslant \epsilon\}\cap M(\gamma,\delta)} \pi(\gamma)d\gamma \\ &\gtrsim \int_{\{||\gamma-\gamma_0||_1 \leqslant \epsilon\}\cap M(\gamma,\delta)} \epsilon^{k_0} \prod_{j=1}^{k_0} g_0(\gamma_j)d\gamma \\ &\gtrsim \int_{N(\gamma,\epsilon/k_0)\cap M(\gamma,\delta_1)} \epsilon^{k_0} \prod_{j=1}^{k_0} g_0(\gamma_j)d\gamma\end{aligned}$$

Now let us show that $N(\gamma, \epsilon/k_0) \subseteq M(\gamma, \delta_1)$. Consider pairs (s, j) with $s \neq j$. Without loss of generality assume $\gamma_{0s} > \gamma_{0j}$. Now, consider the possible values of (γ_j, γ_s) contained in the set $N(\gamma, \epsilon/k_0)$. The smallest distance between values of γ_s and γ_j contained in $N(\gamma, \epsilon/k_0)$ is

$$(\gamma_{0s} - \epsilon/k_0) - (\gamma_{0j} + \epsilon/k_0) \geqslant \epsilon_1 - 2\epsilon/k_0 \geqslant \epsilon_1(1 - 1/k_0) = \delta_1$$

Since the previous holds for any pair (s, j) with $s \neq j$, we have $N(\gamma, \epsilon/k_0) \subseteq M(\gamma, \delta_1)$. Therefore,

$$\begin{aligned}\pi(||\gamma-\gamma_0||_1 \leqslant \epsilon) &\gtrsim \int_{N(\gamma,\epsilon/k_0)} \epsilon^{k_0} \prod_{j=1}^{k_0} g_0(\gamma_j)d\gamma \\ &\gtrsim \epsilon^{k_0} \exp\{-g_1 k_0 \log(1/\epsilon)\} \\ &\gtrsim \exp\{-(g_1+1)k_0 \log(1/\epsilon)\}\end{aligned}$$

for a constant $g_1 > 0$. □

Proof of theorem 6. Only for this proof and for ease of notation the density f will be referred as f_θ. Define the non identifiability set as $T = \{\theta : f_\theta = f_0\}$. In order to define each vector in T, let $0 = t_0 < t_1 < t_2 \ldots < t_{k_0} \leqslant k$ and $\gamma_j = \gamma_{0i}$ for $j \in I_i = \{t_{i-1}+1, t_i\}$. Let $p_{0i} = \sum_{j=t_{i-1}+1}^{t_i} p_j$ and $p_j = 0$ for $j > t_{k_0}$. Define $q_j = p_j/p_{0i}$ for $j \in I_i$. Define $A_n = \left\{\min_{\sigma\in S_k}\left(\sum_{i=1}^{k-k_0} p_{\sigma(i)}\right) > \delta_n M_n\right\}$ and $A'_n = A_n \cap \{\|f - f_0\|_1 \leqslant M\delta_n\}$. Let $D_n = \int_{\{\|f-f_0\|_1<\delta_n\}} \exp(l_n(\theta) - l_n(\theta_0))d(\pi \times \lambda)(\theta)$ with $l_n(\theta_0)$ being the log-likelihood evaluated at θ_0. Along the line of Rousseau and Mengersen (2011)'s proof, to prove theorem 1 we need to show that for any $\epsilon > 0$ there are positive constants

m_1, m_2 and a permutation $\sigma \in S_k$ such that

$$D_n \geqslant m_1 n^{-s(k_0,\alpha)/2} \tag{A.4}$$

$$\Pi(A_n') \leqslant m_2 \delta_n^{s(k_0,\alpha)} M_n^{\bar{\alpha}-m/2-r_2} \tag{A.5}$$

with $s(k_0, \alpha) = k_0 - 1 + mk_0 + \sum_{j=1}^{k-k_0} \alpha_{\sigma(j)}$. Following Rousseau and Mengersen (2011)'s proof, we can show that, under condition B5, (A.4) is satisfied for sufficiently large n. Concerning (A.5), Rousseau and Mengersen (2011) showed that on A_n', there is a set I_i containing indices j_1 and j_2 such that

$$|\gamma_{j_1} - \gamma_{0i}| \leqslant (\delta_n / q_{j_1})^{1/2}, \quad |\gamma_{j_2} - \gamma_{0i}| \leqslant (\delta_n / q_{j_2})^{1/2}$$

with $q_{j_1} > \epsilon/k_0$ and $q_{j_2} > \delta_n M_n/2$. The triangle inequality implies

$$|\gamma_{j_1} - \gamma_{j_2}| \leqslant 2 \left\{\delta_n / \min(q_{j_1}, q_{j_2})\right\}^{1/2}$$

Now, for sufficiently large n, $\min(q_{j_1}, q_{j_2}) > \delta_n M_n/2$ and therefore $|\gamma_{j_1} - \gamma_{j_2}| \lesssim M_n^{-1/2}$. Since g is bounded above by a positive constant, it exists a constant $c > 0$ such that

$$h(\gamma) \leqslant cg \left\{d(\gamma_{j_1}, \gamma_{j_2})\right\} \leqslant cg \left(M_n^{-1/2}\right) \tag{A.6}$$

for $\gamma \in A'$. Let the prior probability of the set A_n' be defined as $\Pi(A_n') = \int_{A_n'} d(\pi \times \lambda)(\gamma \times p)$. To find an upper bound for this integral, directly apply the proof of Rousseau and Mengersen (2011) showing that $\Pi(A_n') \leqslant g\left(M_n^{-1/2}\right) \delta_n^{s(k_0,\alpha)} M_n^{\bar{\alpha}-m/2}$. By assumption, for sufficiently large n, $g\left(M_n^{-1/2}\right) \leqslant r_1 M_n^{-r_2}$. Letting $s_{r_2} = r_2 + m/2 - \bar{\alpha}$, it follows

$$\Pi(A_n') \leqslant M_n^{-s_{r_2}} \delta_n^{s(k_0,\alpha)}$$

□

Appendix B

Chapter 2: Additional Results

B.1 Synthetic examples

Densities in figure 2.2 were defined as follows. Density (Ia) is a standard normal density, density (Ib) is a two components mixture of Gaussians with weights $(0.7, 0.3)$, location parameters $(0, 0)$ and scale parameters $(0.2, 2)$. Density (Ic) is a Students t density with eight degrees of freedom. Density (IIa) is a two-components mixture of Gaussians with mixture weights $(0.3, 0.7)$, location parameters $(-0.8, 0.8)$ and variances $(0.2, 0.2)$. Density (IIb) is a mixture having the same weights and scale parameters as density (IIa) but location parameters $(-1.5, 1.5)$, resulting in better separated clusters. Density $(IIIa)$ is a mixture of a Gaussian with mean 0.7, variance 0.2 and weight 0.7 and a Pearson density with mean -0.7, variance 0.2, weight 0.3, skewness parameter -0.5 and kurtosis parameter 3. Density $(IIIb)$ is a mixture having the same weights, scale parameters, skewness and kurtosis parameters as density $(IIIa)$ but having location parameter $(-1.2, 1.2)$, resulting in better separated clusters. Density (IV) is a bivariate mixture of two Gaussians with weight 0.5, location parameters $(0, 0)$ and $(2, 1)$, variances $(0.2, 0.2)$ and $(0.1, 0.1)$ and correlation

coefficients 0.7 and 0.

Hyperparameters (a_σ, b_σ) for the density of the scale parameter were set to $(3, 1)$. Parameters α_js were all set equal to the same value $\tilde{\alpha}$ and in accordance with Rousseau and Mengersen (2011)'s specification for the density of the weights. For the non-repulsive model, the kernel locations were given independent standard normal priors. For the repulsive model, we considered a repulsion function defined as (4), with g defined as (5) and we chose g_0 to be the standard normal. The distance involved in the repulsion function was chosen to be the symmetric K-L divergence for repulsive priors satisfying definition 1(i) and the Euclidean distance for repulsive priors satisfying definition 1(ii). We chose parameters τ as described in §2.2.2. In particular, the separation level c used to calibrate τ was fixed at six.

Section 2.3 presents results such as misclassification errors and K-L divergences. These quantities were derived as follows. The misclassification error was calculated based on the posterior similarity matrix. Letting n be the number of observations, the similarity matrix is defined as a n-dimensional square matrix with (i, j) element equal to one if the ith and the jth observation belong to the same group and zero otherwise. Let S be the true similarity matrix and $\hat{S}_h$ be the similarity matrix obtained at the hth Markov chain Monte Carlo iteration. Let $S(i, j)$ be the (i, j) element of the matrix S and define the misclassification error m_h as

$$m_h = \frac{1}{n_p} \sum_{i=1}^{n} \sum_{j=i+1}^{n} 1\left(\hat{S}_h(i, j) \neq S(i, j)\right)$$

with n_p being the number of distinct pairs in which n observations may be combined and $1(\cdot)$ the indicator function. The approximation of the K-L divergence at the hth iteration was calculated through

$$kl_h = \sum_{j=1}^{s} \log\left\{f_0(y_{0j})/f(y_{0j}; \theta_h)\right\}$$

with f_0 being the true density, f the fitted density, θ_h the posterior sample at the hth iteration of parameters involved in f and $y_0 = \{y_{01}, \ldots, y_{0s}\}$ being s draws from the true density f_0. In all the experiments s was chosen to be $10,000$.

B.2 Additional results

As mentioned in §2.3, knowing that the smoothing parameter $\tilde{\alpha}$ directly affects the behavior of the mixture weights, it might be argued that under an accurate choice of $\tilde{\alpha}$, the non-repulsive prior may perform as well as the repulsive prior in emptying the extra components. Hence, we ran the non-repulsive model for different values of $\tilde{\alpha}$. This comparison was done by utilizing $1,000$ draws from density IIb. The upper bound on the number of components was chosen to be six and the repulsive prior was chosen to satisfy definition 1(ii). The slice sampler was run for 10, 000 iterations with a burn-in of 5, 000. The chain was thinned by keeping every 10th draw. Table B.1 provides posterior summary statistics for parameters involved in the repulsive model and non-repulsive model for different choices of $\tilde{\alpha}$. Clearly, as $\tilde{\alpha}$ decreases, the non-repulsive model empties the extra components. However, we also see that the 95% credible interval of the location parameters now does not include the true value. This might be explained by the fact that as lower values of $\tilde{\alpha}$ are considered, the posterior can concentrate on too few components leading to degenerate results in terms of estimates of specific component parameters.

Table B.2 shows extra components weights and K-L divergence for datasets drawn from density (IIa, IIb) under repulsive and non-repulsive atoms with six and ten components. As the number of components increases, the probability weight on the extra components remains close to zero under repulsive mixture priors, while the probability weight can grow substantially under non-repulsive priors. Hence, the degraded performance in clustering reported for non-repulsive mixtures relative to repulsive mixtures for the $k = 6$ case becomes more pronounced in the $k = 10$ case.

Table B.1: Percentiles 2.5th and 97.5th of sum of extra weights (s_{ew}) and location parameters involved in the two components with highest weights (μ_1, μ_2) under repulsive and non-repulsive atoms for different values of $\tilde{\alpha}$ considering $1,000$ draws from density IIb

				N-R				R	
$\tilde{\alpha}$		1/3		1/10		1/100		1/3	
	true	2.5%	97.5%	2.5%	97.5%	2.5%	97.5%	2.5%	97.5%
μ_1	1.50	1.51	1.54	1.52	1.53	1.52	1.52	1.49	1.52
μ_2	−1.50	−1.51	−1.44	−1.50	−1.47	−1.49	−1.47	−1.54	−1.43
s_{ew}	0.00	0.00	0.14	0.00	0.02	0.00	0.01	0.00	0.01

Concerning the estimation performance, the K-L divergences resulting from repulsive and non-repulsive mixtures are very similar for high sample sizes.

Table B.2: Mean and standard deviations of the total probability weight placed on extra components (more than used in generating data) and K-L divergence under non-repulsive and repulsive mixtures in different synthetic data cases.

	k=6				k=10			
Data	IIa		IIb		IIa		IIb	
Model	N-R	R	N-R	R	N-R	R	N-R	R
Extra weights								
$n = 100$	0.21	0.08	0.09	0.02	0.34	0.23	0.15	0.06
	(0.11)	(0.07)	(0.07)	(0.02)	(0.11)	(0.09)	(0.09)	(0.04)
$n = 1000$	0.21	0.09	0.03	0.00	0.32	0.15	0.05	0.01
	(0.11)	(0.06)	(0.04)	(0.01)	(0.11)	(0.08)	(0.04)	(0.01)
K-L								
$n = 100$	0.03	0.05	0.07	0.08	0.03	0.06	0.08	0.10
	(0.01)	(0.02)	(0.02)	(0.03)	(0.02)	(0.03)	(0.02)	(0.04)
$n = 1000$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	(0.00)	(0.00)	(0.03)	(0.03)	(0.00)	(0.00)	(0.00)	(0.00)

The value of τ in the repulsive prior 2.4 relies upon the choice of the separation level c. In order to assess the sensitivity of results to this choice, the K-L divergence was computed for different separation levels. For this comparison, observations were drawn from densities (IIa) and (IIb). Mixtures of six and ten components were

fitted using a Gibbs sampler. The slice sampler was run for $10,000$ iterations with a burn-in of $5,000$. The chain was thinned by keeping every 10th draw. Figure B.1 shows the median of the K-L divergence between the true and the estimated density. Clearly, as the separation level increases, the K-L divergence remains stable.

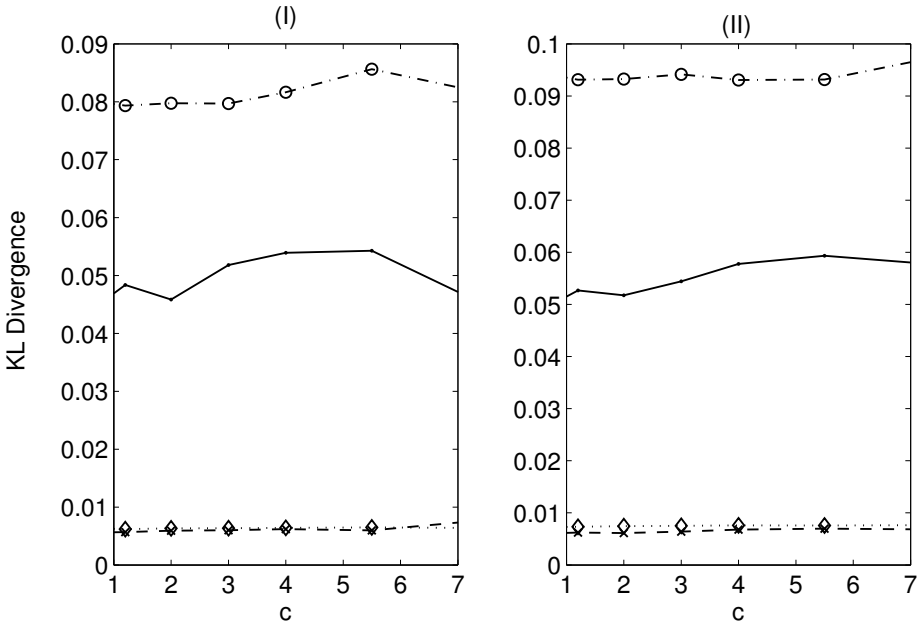

FIGURE B.1: Plot of K-L divergence under six and ten components $(6 : I, 10 : II)$ for different choice of separation level c under density (IIa) for different sample sizes (100:solid ; 1000:dash) and density (IIb) for different sample sizes (100:dash-dot; 1000:dot)

Bibliography

Aguilar, O. and West, M. (2000), "Bayesian dynamic factor models and portfolio allocation," *Journal of Business and Economic Statistics*, 18, 338–357.

Allard, W., Chen, G., and Maggioni, M. (2012), "Multiscale geometric methods for data sets II: geometric wavelets," *Applied and Computational Harmonic Analysis*, 32, 435–462.

Arden, R., Chavez, R. S., Grazioplene, R., and Jung, R. E. (2010), "Neuroimaging creativity: a psychometric view." *Behavioural brain research*, 214, 143–156.

Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M. (2003), "Bayesian factor regression models in the large p small n paradigm," *Bayesian Statistics*, 7, 733–742.

Bhattacharya, A. and Dunson, D. B. (2011), "Sparse Bayesian infinite factor models," *Biometrika*, 98, 291–306.

Bishop, C. M. and Svensen, M. (2003), "Bayesian hierarchical mixtures of experts," *Nineteenth Conference on Uncertainty in Artificial Intelligence* , pp. 57–64.

Breiman, L. (1996), "Bagging predictors," *Machine Learning*, 24, 123140.

Breiman, L. (2001), "Random forests," *Machine Learning*, 45, 5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984), *Classification and regression trees*, Chapman & Hall/CRC.

Chauveau, D. and Diebolt, J. (1998), "An automated stopping rule for MCMC convergence assessment," *Computational Statistics*, 14, 419–442.

Chen, M., Silva, J., and Paisley, J. (2010), "Compressive Sensing on Manifolds Using a Nonparametric Mixture of Factor Analyzers: Algorithm and Performance Bounds," *IEEE TRANSACTIONS ON SIGNAL PROCESSING* , 58, 6140–6155.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1993), "Bayesian CART model search," *Journal of the American Statistical Association*, 443, 935–948.

Chiu, T., Leonard, T., and Tsui, K. (1996), "The matrix-logarithmic covariance model," *Journal of the American Statistical Association*, 91, 198–210.

Chung, Y. and Dunson, D. B. (2009), "Nonparametric Bayes conditional distribution modeling with variable selection," *Journal of the American Statistical Association*, 104, 1646–1660.

Coifman, R. and Lafon, S. (2006), "Diffusion Maps," *Applied and Computational Harmonic Analysis*, 21, 5–30.

Cron, A. J. and West, M. (2011), "Efficient Classification-Based Relabeling in Mixture Models," *The American Statistician*, 65, 16–20.

Daley, D. J. and Vere-Jones, D. (2008), *An Introduction to the Theory of Point Processes*, Springer.

Dasgupta, S. (1999), "Learning Mixtures of Gaussians," *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pp. 633–644.

Dasgupta, S. and Schulman, L. (2007), "A Probabilistic Analysis of EM for Mixtures of Separated, Spherical Gaussians," *The Journal of Machine Learning Research*, 8, 203–226.

Davis, D. T. and Hwang, J. N. (1998), "Expanding Gaussian Kernels for Multivariate Conditional Density Estimation," *IEEE Transactions on Signal Processing*, 46, 269–275.

Death, G. (2002), "Multivariate Regression Trees: A New Technique for Modeling Species-Environment Relationships," *Ecology*, 83, 1105–1117.

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, 39, 1–38.

Dunson, D. B., Pillai, N., and Park, J. H. (2007), "Bayesian density regression," *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 69, 163–183.

Escobar, M. D. and West, M. (1995), "Bayesian Density Estimation and Inference Using Mixtures," *Journal of the American Statistical Association*, 90, 577–588.

Fan, J. Q. and Yim, T. H. (2004), "A crossvalidation method for estimating conditional densities," *Biometrika*, 91, 819–834.

Fan, J. Q., Yao, Q. W., and Tong, H. (1996), "Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems," *Biometrika*, 83, 189–206.

Figueiredo, M. A. T. and Jain, A. K. (2002), "Unsupervised Learning of Finite Mixture Models," *IEEE transactions on pattern analysis and machine intelligence*, 24, 381–396.

Fraley, C. and Raftery, A. E. (2002), "Model-Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, 97, 611–631.

Fu, G., Shih, F. Y., and Wang, H. (2011), "A kernel-based parametric method for conditional density estimation," *Pattern recognition*, 44, 284–294.

Fyshe, A., Fox, E., and Dunson, D. (2012), "Hierarchical Latent Dictionaries for Models of Brain Activation," *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Gelfand, A., Schmidt, A., Banerjee, S., and Sirmans, C. (2004), "Nonstationary multivariate process modeling through spatially varying coregionalization," *Test*, 13, 263–312.

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

Ghahramani, Z. and Beal, M. J. (2000), "Variational inference for Bayesian mixtures of factor analyzers," *Neural Information Processing Systems 12*, 52, 449–455.

Ghahramani, Z. and Hinton, G. E. (1997), "The EM algorithm for factor analyzers," *Technical Report Number CRG-TR-96-1, The University of Toronto, Toronto*.

Ghosal, S., Ghosh, J. K., and Van Der Vaart, A. W. (2000), "Convergence Rates of Posterior Distributions," *The Annals of Statistics*, 28, 500–531.

Gray, W. R., Bogovic, J. A., Vogelstein, J. T., Landman, B. A., Prince, J. L., and Vogelstein, R. J. (2010), "Magnetic resonance connectome automated pipeline: an overview." *IEEE pulse*, 3, 42–8.

Griffin, J. E. and Steel, M. F. J. (2006), "Order-based dependent Dirichlet processes," *Journal of the American Statistical Association*, 101, 179–194.

Hannah, L. A., Blei, D. M., and Powell, W. B. (2011), "Dirichlet Process Mixtures of Generalized Linear Models," *Journal of Machine Learning Research*, 1, 1–33.

Hastie, D. and Green, P. J. (2012), "Model choice using reversible jump Markov chain Monte Carlo," *Statistica Neerlandica*, 66, 309–338.

Hatjispyros, S. J., Nicoleris, T., and Walker, S. G. (2011), "Dependent mixtures of Dirichlet processes," *Computational Statistics & Data Analysis*, 55, 2011–2025.

Hoff, P. D. and Niu, X. (2012), "A Covariance Regression Model," *Statistica Sinica*, 22, 729–753.

Holmes, M. P., Gray, G. A., and Isbell, C. L. (2010), "Fast kernel conditional density estimation: a dual-tree Monte Carlo approach," *Computational statistics & data analysis*, 54, 1707–1718.

Hothorn, T., Hornik, K., and Zeileis, A. (2006), "Unbiased Recursive Partitioning: A Conditional Inference Framework," *Journal of Computational and Graphical Statistics*, 15, 651–674.

Huber, M. L. and Wolpert, R. L. (2009), "Likelihood-Based Inference for Matern Type-III Repulsive Point Processes," *Advances in Applied Probability*, 41, 958–977.

Inouye, M., Kettunen, J., Soininen, P., Silander, K., Ripatti, S., Kumpula, L. S., Hamalainen, E., Jousilahti, P., Kangas, A. J., Mannisto, S., Savolainen, M. J., Jula, A., Leiviska, J., Palotie, A., Salomaa, V., Perola, M., Ala-Korpela, M., and Peltonen, L. (2010), "Metabonomic, transcriptomic, and genomic variation of a population cohort," *Molecular Systems Biology*, 6, 1744–4292.

Ishwaran, H. and James, L. F. (2001), "Gibbs Sampling Methods for Stick-Breaking Priors," *Journal of the American Statistical Association*, 96, 161–173.

Ishwaran, H. and Zarepour, M. (2002), "Dirichlet Prior Sieves in Finite Normal Mixtures," *Statistica Sinica*, 12, 941–963.

Ishwaran, H., James, L. F., and Sun, J. (2001), "Bayesian Model Selection in Finite Mixtures by Marginal Density Decompositions," *Journal of American Statistical Association*, 96, 1316–1332.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991), "Adaptive mixture of local experts," *Neural Computation*, 3, 79–87.

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005), "MCMC and the Label Switching Problem in Bayesian Mixture Models," *Statistical Science*, 20, 50–67.

Jiang, W. X. and Tanner, M. A. (1999), "Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation," *Annals of Statistics*, 27, 987–1011.

Jordan, M. I. and Jacobs, R. A. (1994), "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, 6, 181–214.

Jung, R. E., Grazioplene, R., Caprihan, A., Chavez, R. S., and Haier, R. J. (2010), "White matter integrity, creativity, and psychopathology: Disentangling constructs with diffusion tensor imaging," *PloS one*, 5, e9818.

Kalli, M., Griffin, J. E., and Walker, S. G. (2011), "Slice Sampling Mixture Models," *Statistics and Computing*, 21, 93–105.

Karypis, G. and Kumar, V. (1999), "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing 20*, 1, 359392.

Krauthausen, P. and Hanebeck, U. D. (2010), "Regularized non-parametric multivariate density and conditional density estimation," *IEEE Conference on Multisensor Fusion and Integration*, pp. 180–186.

Larsen, D. R. and Speckman, P. L. (2004), "Multivariate Regression Trees for Analysis of Abundance Data," *Biometrics*, 60, 543–549.

Lavine, M. and West, M. (1992), "A Bayesian Method for Classification and Discrimination," *Canadian Journal of Statistics*, 20, 451–461.

Lawson, A. and Clark, A. (2002), *Spatial Cluster Modeling*, Chapman & Hall CRC, London, UK.

Locarek-Junge, H. and Weihs, C. (2009), *Classification as a Tool for Research*, Springer.

Lopes, H. F. and West, M. (2004), "Bayesian model assessment in factor analysis," *Statistica Sinica*, 14, 41–67.

Lopes, H. F., Gamerman, D., and Salazar, E. (2011), "Generalized spatial dynamic factor models," *Computational Statistics & Data Analysis*, 55, 1319–1330.

Lutz, R. W. and Buhlmann, P. (2006), "Boosting for high multivariate responses in high dimensional linear regression," *Statistica Sinica*, 16, 471–494.

Mallick, B. K. (1998), "Bayesian CART model search," *Biometrika*, 85, 363–377.

McLachlan, G. J. and Peel, D. (2000), *Finite Mixture Models*, vol. 299, Wiley Series in Probability and Statistics.

Meeds, E. and Osindero, S. (2006), "Bayesian hierarchical mixtures of experts," *Advances in Neural Information Processing Systems*.

Minka, T. (2001), "Automatic choice of dimensionality for PCA," *Advances in neural information processing systems*, pp. 598–604.

Mori, S. and Zhang, J. (2006), "Principles of diffusion tensor imaging and its applications to basic neuroscience research." *Neuron*, 51, 527–39.

Mossavat, I. and Amft, O. (2011), "Sparse bayesian hierarchical mixture of experts," *IEEE Statistical Signal Processing Workshop (SSP)*.

Muller, P., Erkanly, A., and West, M. (1996), "Bayesian curve tting using multivariate normal mixtures," *Biometrika*, 83, 67–79.

Muthen, B. and Shedden, K. (1999), "Finite Mixture Modeling with Mixture Outcomes Using the EM Algorithm," *Biometrics*, 55, 463–469.

Neal, R. M. (2003), "Slice Sampling," *The Annals of Statistics*, 31, 705–767.

Norets, A. and Pelenis, J. (2012), "Bayesian modeling of joint and conditional distributions," *Journal of Econometrics*, 168, 332–346.

Nott, D. J., Tan, S. L., Villani, M., and Kohn, R. (2012), "Regression density estimation with variational methods and stochastic approximation," *Journal of Computational and Graphical Statistics*, 21, 797–820.

Onatski, A. (2005), "Determining the number of factors from empirical distribution of eigenvalues," *The Review of Economics and Statistics*, 9, 557–587.

Pourahmadi, M. (1999), "Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation," *Biometrika*, 86, 677–690.

Power, J. D., Barnes, K. A., Stone, C. J., and Olshen, R. A. (2012), "Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion," *Neuroimage*, 59, 2142–2154.

Raftery, A. E. and Fraley, C. (1998), "How Many Clusters? Which Clustering Method? Answers via Model-Based Cluster Analysis," *The Computer Journal*, 41, 578–588.

Rahman, I. U., Drori, I., Stodden, V. C., and Donoho, D. L. (2005), "Multiscale representations for manifold- valued data," *SIAM J. Multiscale Model*, 4, 1201–1232.

Rasmussen, C. E. and Ghahramani, Z. (2002), "Infinite mixtures of Gaussian process experts," *Advances in neural information processing systems 14*.

Ren, L., Du, L., Carin, L., and Dunson, D. B. (2011), "Logistic stick-breaking process," *Journal of Machine Learning Research*, 12, 203–239.

Richardson, S. and Green, P. (1997), "On Bayesian Analysis of Mixtures with an Unknown Number of Components," *Journal of the Royal Statistical Society B*, 59, 731–758.

Roberts, G. O. and Rosenthal, J. S. (2007), "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms," *Journal of Applied Probability*, 44, 458–475.

Rodriguez, A. and Dunson, D. B. (2011), "Nonparametric Bayesian models through probit stick-breaking processes," *Bayesian Analysis*, pp. 145–178.

Rousseau, J. and Mengersen, K. (2011), "Asymptotic Behaviour of the Posterior Distribution in Over-Fitted Models," *Journal of the Royal Statistical Society B*, 73, 689–710.

Scricciolo, C. (2011), "Posterior Rates of Convergence for Dirichlet Mixtures of Exponential Power Densities," *Electronic Journal of Statistics*, 5, 270–308.

Seber, G. A. F. (2004), *Multivariate observations*, Wiley.

Sethuraman, J. (1994a), "A Constructive Denition of Dirichlet Priors," *Statistica Sinica*, 4, 639–650.

Sethuraman, J. (1994b), "A constructive denition of Dirichlet priors," *Statistica Sinica*, 4, 639–650.

Shahbaba, B. and Neal, R. (2009), "Non linear models using Dirichlet process mixtures," *Journal of Machine Learning Research*, 10, 1829–1850.

Shapire, R., Freund, Y., Bartlett, P., and Lee, W. (1998), "Boosting the margin: a new explanation for the effectiveness of voting methods," *Annals of Statistics*, 26, 16511686.

Sikka, S., Vogelstein, J. T., and Milham, M. P. (2012), "Towards Automated Analysis of Connectomes: The Configurable Pipeline for the Analysis of Connectomes (C-PAC)," in *Organization of Human Brain Mapping*, Neuroinformatics.

Stephens, M. (2000a), "Bayesian Analysis of Mixture Models with an Unknown Number of Components - An Alternative to Reversible Jump Methods," *The Annals of Statistics*, 28, 40–74.

Stephens, M. (2000b), "Dealing with label switching in mixture models," *Journal of the Roya; statistical society B*, 62, 795–810.

Sugar, C. and James, G. (2003), "Finding the number of clusters in a data set: an information theoretic approach," *Journal of the American Statistical Association*, 98, 750–763.

Tipping, M. E. and Bishop, C. M. (1997), "Mixtures of Principal Component Analysers," *In Proceedings IEE Fifth International Conference on Articial Neural Networks*, pp. 13–18.

Tipping, M. E. and Bishop, C. M. (2012), "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, 61, 611–622.

Tokdar, S. T., Zhu, Y. M., and Ghosh, J. K. (2010), "Bayesian density regression with logistic Gaussian process and subspace projection," *Bayesian Analysis*, 5, 319–344.

Tran, M. N., Nott, D. J., and Kohn, R. (2012), "Simultaneous variable selection and component selection for regression density estimation with mixtures of heteroscedastic experts," *Electronic Journal of Statistics*, 6, 1170–1199.

Utsugi, A. and Kumagai, T. (2011), "Bayesian analysis of mixtures of factor analyzers," *Neural Computation*, 13, 993–1002.

Wang, J. (2010), "Consistent selection of the number of clusters via crossvalidation," *Biometrika*, 97, 893–904.

Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelic, A., von Rohr, P., Thiele, L., et al. (2004), "Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana," *Genome Biol*, 5, R92.

Williams, P. M. (1996), "Using Neural Networks to Model Conditional Multivariate Densities," *Neural Computation*, 8, 843–854.

Wu, Y., Tjelmeland, H., and West, M. (2007), "Bayesian CART: Prior Specification and Posterior Simulation," *Journal of Computational and Graphical Statistics*, 16, 44–66.

Yao, W. and Lindsay, B. G. (2009), "Bayesian mixture labeling by highest posterior density," *Journal of the American Statistical Association*, 104.

Zhou, X. and Liu, X. (2008), "The EM algorithm for the extended finite mixture of the factor analyzers model," *Computational Statistics and Data Analysis*, 52, 3939–3953.

Zou, Q.-H., Zhu, C.-Z., Yang, Y., Zuo, X.-N., Long, X.-Y., Cao, Q.-J., Wang, Y.-F., and Zang, Y.-F. (2008), "An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: fractional ALFF." *Journal of neuroscience methods*, 172, 137–141.

Biography

Francesca Petralia was born in Vigevano, Italy on July 5, 1983. She graduated with honors from the University of Pavia with a Bachelor of Science in Economics in June 2006. From the same university she earned her Master of Science in Finance in June 2008. She then joined the Department of Statistical Science at Duke University to attend the Ph.D. program. In May 2012 she earned a Master of Science in Statistics.