Particle Learning for General Mixtures

By CARLOS M. CARVALHO
carlos.carvalho@mccombs.utexas.edu
HEDIBERT F. LOPES
hedibert.lopes@chicagobooth.edu
NICHOLAS G. POLSON
nicholas.polson@chicagobooth.edu
and MATTHEW A. TADDY
matt.taddy@chicagobooth.edu

McCombs School of Business, University of Texas at Austin,
Austin, Texas 78712, U.S.A.
and
Booth School of Business, University of Chicago,
Chicago, Illinois 60637, U.S.A.

Abstract

This paper develops particle learning (PL) methods for the estimation of general mixture models. The approach is distinguished from alternative particle filtering methods in two major ways. First, each iteration begins by resampling particles according to posterior predictive probability, leading to a more efficient set for propagation. Second, each particle tracks only the "essential state vector" thus leading to reduced dimensional inference. In addition, we describe how the approach will apply to more general mixture models of current interest in the literature; it is hoped that this will inspire a greater number of researchers to adopt sequential Monte Carlo methods for fitting their sophisticated mixture based models. Finally, we show that PL leads to straightforward tools for marginal likelihood calculation and posterior cluster allocation.

Key Words: Nonparametric, mixture Models, EM, MCMC, variational methods, particle filtering, learning, stochastic simulation, Bayesian Inference, Dirichlet process, Indian buffet process, probit stick-breaking.

1 Introduction

Mixture models provide a flexible and intuitive framework for inference. The practical utility of such models is derived from separation of the full model, conditional on latent allocation of observations to mixture components, into a set of distribution functions that are analytically manageable and of a standard form. In addition, the predictive probability function is, even for models with an infinite number of mixture components, often available and offers a workable sampling model for both new observations and the associated latent mixture allocation.

Many different methods are available for fitting general mixture models. Commonly used approaches are EM-type algorithms, Markov chain Monte Carlo (MCMC) and variational Bayes (VB) optimization. Examples in the literature include, among others, Escobar and West (1995), Richardson and Green (1997), MacEachern and Müller (1998), Stephens (2000), and Blei and Jordan (2006). Additional sampling alternatives appear in Chopin (2002),Del Moral, Doucet and Jasra (2006), Jasra, Stephen and Holmes (2007) and Fearnhead and Meligkotsidou (2007). The relative merit of these techniques is dependent on the data analysis setting of interest: EM will find modal clusters fairly efficiently, but is difficult in high dimensions and leads only to point estimates. MCMC is able to provide a complete picture of the posterior distribution, but the time-consuming Markov Chain must be re-run to update for every addition to the batch of observations. Finally, variational Bayes methods are robust in high-dimensional large dataset applications, but provide only an approximation to the posterior that is of unknown quality.

Without attempting to supplant any of these existing techniques in their respective strengths, this article develops an efficient framework for *sequential* sampling from the posterior distribution for general mixture models. Advantages of such an approach in on-line inference settings include: *(i)* uncertainty is updated practically instantly as new information arrives, *(ii)* a filtering process makes explicit the information gained as time progresses, and *(iii)* marginal data likelihood estimates are easily obtained from sequential predictive densities. In particular, we adapt the particle learning (PL) approach of Carvalho, Johannes, Lopes, and Polson (2010) and Lopes, Carvalho, Johannes and Polson (2010) to general mixtures. A general framework is introduced for applying the PL resample/propagate procedure at each "time" point (i.e., given a new observation) to provide a sequential particle approximation to the posterior. Use

of the proposed methodology is then illustrated in density estimation problems, as well as for latent feature and dependent nonparametric mixture models.

Ours is not the first application of sequential Monte Carlo sampling to mixture models. The most sophisticated methodology can be cast within a general particle filtering framework presented by MacEachern, Clyde, and Liu (MCL; 1999), which builds on earlier work by Kong, Liu, and Wong (1994). Although other authors have suggested improvements (e.g., Fearnhead (2004) describes a more efficient sampling step), MCL algorithms form the basis for the existing state of the art. The distinction is subtle between their framework and the PL approach introduced herein; indeed, MCL algorithm S2 involves the same probabilistic calculations as the corresponding application of our PL approach, with major differences only in the order of operations and what information is tracked as a "particle". However, despite demonstration by MacEachern *et al* of the utility of these algorithms, the methodology has not been as widely adopted and applied as one would expect. This could be due in some part to presentation of the material: the general methods are derived in terms of a somewhat narrow class of mixture models, and structural advantages of the most useful algorithms are perhaps not highlighted in such a way as to ease application in alternative settings. But, at the same time, the PL approach is able to alleviate some real practical limitations of MCL; the way that this is achieved can be explained through two major differences between the frameworks.

The first difference is related to pre-selection of particles. The MCL particle method utilizes a standard sequential importance sampling approach, where the propagation of states is carried out before an importance reweighting of particles. In contrast, PL always resamples particles first, proportional to the predictive probability of a new observation. In this regard, PL can be interpreted as an Auxiliary Particle Filter (Pitt and Shephard, 1999) version of MCL. Liu and Chen (1998) discuss the potential advantages of re-sampling first in the context of dynamic systems where only state variables are unknown – we take this discussion further by explicitly applying it to general mixtures models where we need to deal with uncertainty about both states and parameters defining the model.

The second difference is related to the information PL tracks over time. The MacEachern *et al.* framework attempts to track a smoothed distribution of latent allocations of each observation to a mixture component. This relies on repeated importance sampling reweightings

of the allocation vector, which has length equal to the present number of observations. The dimension of this target of inference grows rapidly in time, leaving the procedure more susceptible to unbalanced weights and particle degeneracy. In contrast, PL tracks only the sufficient information for filtering defined by the essential state vector that allows for the computation of the resampling and propagating steps. Observation allocation (and the implied clustering) is deferred to the end of the filtering process and it is obtained through a backwards particle smoothing algorithm.

In addition to presenting an algorithm that is able to improve upon MCL in these two ways, this article aims to show applicability of the PL framework to a very general class of mixture models. We thus provide, in Section 1, a generic formulation of the PL approach to a broad class of mixture specifications, beyond the standard finite or Dirichlet process mixture models. Section 2 details algorithms for density estimation through both finite (2.1) and infinite (2.2) mixture models, including the widely used Dirichlet process mixture model. Section 3 derives algorithms for two alternative applications of nonparametric mixture-based models: latent feature modeling through the Indian buffet process (3.1) and probit stick-breaking models for dependent random measures (3.2). The list of applications described herein is not meant to be exhaustive but rather to provide a set of examples that demonstrate the applicability and generality of PL. Since the steps required by our generic approach are fairly simple and intuitive, this should facilitate the wider use of sequential particle methods in fitting general mixtures. Finally, Section 4 contains three examples of the PL mixtures algorithm applied to simulated data. Section 4.1 involves a simple finite mixture model of Poisson densities, and uses the PL technique for marginal likelihood estimation to determine the number of mixture components. Section 4.2 presents density estimation through a Dirichlet process mixture of multivariate normals, with the PL fit algorithm applied to up to 12500 observations of 25 dimensional data. The multivariate normal mixtures are extended to nonparametric regression in Section 4.3, where we illustrate sequential learning for the full underlying random mixing distribution. The article concludes in Section 5 with summary and discussion of the contributions of this PL mixtures framework.

1.1 The General PL Framework for Mixtures

The complete class of mixture models under consideration is defined by the likelihood $\mathrm{p}(y_{t+1}|k_{t+1},\theta)$, a transition equation $\mathrm{p}(k_{t+1}|k^t,\theta)$ with $k^t=\{k_1,\ldots,k_t\}$, and parameter prior $\mathrm{p}(\theta)$. This general formulation can be naturally represented as a state-space model of the following form

$$y_{t+1} = \mathrm{f}(k_{t+1},\theta) \tag{1}$$

$$k_{t+1} = \mathrm{g}(k^t,\theta) \tag{2}$$

where (1) is the observation equation and (2) is the evolution for states k_{t+1}. Note that this structure establishes a direct link to the general class of hidden Markov models, which encompasses a vast number of widely used models. In this, the k_t states refer to a latent allocation of observations to mixture components.

In order to describe the PL algorithm, we begin by defining $\mathcal{Z}_t$ as an "essential state vector" that will be tracked in time. Assume that this vector is sufficient for sequential inference; that is, it allows for the computation of:

(a) the posterior predictive $\mathrm{p}(y_{t+1}|\mathcal{Z}_t)$,

(b) the posterior updating rule $\mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t,y_{t+1})$,

(c) and parameter learning via $\mathrm{p}(\theta|\mathcal{Z}_{t+1})$.

Given an equally weighted particle set $\left\{\mathcal{Z}_t^{(i)}\right\}_{i=1}^N$ which serves to approximate the posterior $\mathrm{p}\left(\mathcal{Z}_t|y^t\right)$, the generic particle learning update for a new observation y_{t+1} proceeds in two steps:

$$\text{Resample } \ \mathcal{Z}_t^{(i)} \propto \mathrm{p}(y_{t+1}|\mathcal{Z}_t^{(i)}) \quad \rightarrow \quad \text{Propagate } \ \mathcal{Z}_{t+1}^{(i)} \sim \mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t^{(i)},y_{t+1}). \tag{3}$$

This process can be understood by re-writing Bayes' theorem as

$$\mathrm{p}\left(\mathcal{Z}_t|y^{t+1}\right) \propto \mathrm{p}\left(y_{t+1}|\mathcal{Z}_t\right)\mathrm{p}\left(\mathcal{Z}_t|y^t\right) \tag{4}$$

$$\mathrm{p}\left(\mathcal{Z}_{t+1}|y^{t+1}\right) = \int \mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t,y_{t+1})d\mathrm{P}\left(\mathcal{Z}_t|y^{t+1}\right), \tag{5}$$

where $\mathrm{P}(\cdot)$ refers throughout to the appropriate continuous/discrete measure. Thus, after resampling the initial particles with weights proportional to $\mathrm{p}(y_{t+1}|\mathcal{Z}_t)$ we have samples from

$\mathrm{p}(\mathcal{Z}_t|y^{t+1})$. These samples are then propagated through $\mathrm{p}(\mathcal{Z}_{t+1}|\mathcal{Z}_t, y_{t+1})$, leading to updated particles $\left\{\mathcal{Z}_{t+1}^{(i)}\right\}_{i=1}^{N}$ approximating $\mathrm{p}(\mathcal{Z}_{t+1}|y^{t+1})$. The method is summarized by the following algorithm.

PL for general mixture models

1. **Resample**: Generate an index $\boldsymbol{\zeta} \sim \mathrm{MN}(\boldsymbol{\omega}, N)$ where

$$\omega(i) = \frac{\mathrm{p}\left(y_{t+1}|\mathcal{Z}_t^{(i)}\right)}{\sum_{i=1}^{N} \mathrm{p}\left(y_{t+1}|\mathcal{Z}_t^{(i)}\right)}$$

2. **Propagate**:

$$\mathcal{Z}_{t+1}^{(\zeta(i))} \sim \mathrm{p}\left(\mathcal{Z}_{t+1}|\mathcal{Z}_t^{(\zeta(i))}, y_{t+1}\right)$$

3. **Learn**:

$$\mathrm{p}(\theta|y^{t+1}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathrm{p}(\theta|\mathcal{Z}_{t+1})$$

The following examples explicitly define the essential state vector and the elements needed to implement the general algorithm described above. These examples are further explore in Section 4.

Example 1: Finite Mixtures of Poisson Densities. An m component mixture of Poisson densities is defined as

$$\mathrm{p}(y_t) = \sum_{i=1}^{m} p_j \mathrm{Po}(y_t; \theta_j^\star), \tag{6}$$

where $\mathbb{E}(y_t|k_t = i) = \theta_i^\star$. We complete the model with conjugate priors $\pi(\theta_j^\star) = \mathrm{ga}(\alpha_j, \beta_j)$, for $j = 1, \ldots, m$, and $\pi(\boldsymbol{p}) \sim \mathrm{Dir}(\boldsymbol{\gamma})$. The form of the conditional posterior given y^t, given the latent allocation k^t, is completely defined by $\mathbf{n}_t$, the number of samples in each component, and sufficient statistics $\mathbf{s}_t = (s_{t,1}, \ldots, s_{t,m})$, where $s_{t,j} = \sum_{r=1}^{t} y_r \mathbb{1}_{[k_r=j]}$. This leads to the following definition of the essential state vector to be tracked in time $\mathcal{Z}_t = \{\mathbf{s_t}, \mathbf{n_t}\}$. We can

then straightforwardly obtain the predictive for y_{t+1} as

$$\begin{aligned}\mathrm{p}(y_{t+1}|\mathcal{Z}_t) &= \sum_{k_{t+1}=j=1}^{m} \int\int p_j \mathrm{p}(y_{t+1}|\theta_j^\star) p(\boldsymbol{\theta}^\star, \boldsymbol{p}) d(\boldsymbol{\theta}^\star, \boldsymbol{p}) \\ &= \sum_{k_{t+1}=j=1}^{m} \frac{\Gamma\left(s_{t,j}+y_{t+1}+\alpha_j\right)}{\Gamma\left(s_{t,j}+\alpha_j\right)} \frac{\left(\beta_j+n_{t,j}\right)^{s_{t,j}+\alpha_j}}{\left(\beta_j+n_{t,j}+1\right)^{s_{t,j}+y_{t+1}+\alpha_j}} \frac{1}{y_{t+1}!} \left(\frac{\gamma_j+n_{t,j}}{\sum_{i=1}^m \gamma_i+n_{t,i}}\right).\end{aligned}$$

Propagating k_{t+1} is done according to

$$p(k_{t+1}=j|\mathcal{Z}_t, y_{t+1}) \propto \frac{\Gamma\left(s_{t,j}+y_{t+1}+\alpha_j\right)}{\Gamma\left(s_{t,j}+\alpha_j\right)} \frac{\left(\beta_j+n_{t,j}\right)^{s_{t,j}+\alpha_j}}{\left(\beta_j+n_{t,j}+1\right)^{s_{t,j}+y_{t+1}+\alpha_j}} \left(\frac{\gamma_j+n_{t,j}}{\sum_{i=1}^m \gamma_j+n_{t,i}}\right).$$

Given k_{t+1}, $\mathcal{Z}_{t+1}$ is updated by the recursions $s_{t+1,j} = s_{t,j} + y_{t+1}\mathbb{1}_{\{k_{t+1}=j\}}$ and $n_{t+1,j} = n_{t,j} + \mathbb{1}_{\{k_{t+1}=j\}}$, for $j = 1, \ldots, m$. Finally, learning about $\theta^\star$ conditional on $\mathcal{Z}_{t+1}$ involves sampling from an update gamma posterior for each particle.

Example 2: DP Mixtures of Multivariate Normals The d-dimensional DP multivariate normal mixture (DP-MVN) model has density function (Escobar and West, 1995)

$$f(y_t; G) = \int \mathrm{N}(y_t|\mu_t, \Sigma_t) dG(\mu_t, \Sigma_t), \quad \text{and} \quad G \sim DP(\alpha, G_0(\mu, \Sigma)), \tag{7}$$

with given concentration parameter α and conjugate centering distribution $G_0 = \mathrm{N}(\mu; \lambda, \Sigma/\kappa)$ $\mathrm{W}(\Sigma^{-1}; \nu, \Omega)$, where $\mathrm{W}(\Sigma^{-1}; \nu, \Omega)$ denotes a Wishart distribution such that $\mathbb{E}[\Sigma^{-1}] = \nu\Omega^{-1}$ and $\mathbb{E}[\Sigma] = (\nu-(d+1)/2)^{-1}\Omega$. This model specification leads to an essential state vector $\mathcal{Z}_t = \{\mathbf{s}_t, \mathbf{n}_t, m_t\}$ where $\mathbf{s}_t$ is the conditional sufficient statistics for each unique mixture component, $\mathbf{n}_t$ is the number of observations assigned to each component and m_t is the number of current components. $\mathbf{s}_t$ is defined by $\bar{y}_{t,j} = \sum_{r:k_r=j} y_r/n_{t,j}$ and $S_{t,j} = \sum_{r:k_r=j}(y_r - \bar{y}_{t,j})(y_r - \bar{y}_{t,j})' = \sum_{r:k_r=j} y_r y_r' - n_{t,j}\bar{y}_{t,j}\bar{y}_{t,j}'$. The predictive density for resampling is

$$\mathrm{p}(y_{t+1}|\mathcal{Z}_t) = \frac{\alpha}{\alpha+t}\mathrm{St}(y_{t+1}; a_0, B_0, c_0) + \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha+t}\mathrm{St}\left(y_{t+1}; a_{t,j}, B_{t,j}, c_{t,j}\right) \tag{8}$$

where the Student's t distributions are parametrized by $a_0 = \lambda$, $B_0 = \frac{2(\kappa+1)}{\kappa c_0}\Omega$, $c_0 = 2\nu - d + 1$, $a_{t,j} = \frac{\kappa\lambda+n_{t,j}\bar{y}_{t,j}}{\kappa+n_{t,j}}$, $B_{t,j} = \frac{2(\kappa+n_{t,j}+1)}{(\kappa+n_{t,j})c_{t,j}}\left[\Omega + \frac{1}{2}D_{t,j}\right]$, $c_{t,j} = 2\nu + n_{t,j} - d + 1$, and $D_{t,j} = S_{t,j} +$

$\frac{\kappa n_{t,j}}{(\kappa+n_{t,j})}(\lambda-\bar{y}_{t,j})(\lambda-\bar{y}_{t,j})'$. Propagating k_{t+1} is done such that

$$\begin{aligned} \text{for } \; j=1,\ldots,m_t, \quad \mathrm{p}(k_{t+1}=j) &\propto \frac{n_{t,j}}{\alpha+t}\mathrm{St}(y_{t+1};\; a_{t,j},B_{t,j},c_{t,j}) \\ \text{and } \; \mathrm{p}(k_{t+1}=m_t+1) &\propto \frac{\alpha}{\alpha+t}\mathrm{St}(y_{t+1};\; a_0,B_0,c_0). \end{aligned} \tag{9}$$

Finally $\mathcal{Z}_{t+1}$ is updated as follows: If $k_{t+1}=m_t+1$, $m_{t+1}=m_t+1$ and $s_{t+1,m_{t+1}}=[y_{t+1},0]$. If $k_{t+1}=j$, $n_{t+1,j}=n_{t,j}+1$, $\bar{y}_{t+1}=(n_{t,j}\bar{y}_{t,j}+y_{t+1})/n_{t+1,j}$ and $S_{t+1,j}=S_{t,j}+y_{t+1}y_{t+1}'+n_{t,j}\bar{y}_{t,j}\bar{y}_{t,j}^{i\prime}-n_{t+1,j}\bar{y}_{t+1,j}\bar{y}_{t+1,j}^{i\prime}$. The remaining sufficient statistics are the same as at time t.

1.2 Allocation

The filtering process for $\mathcal{Z}_t$ does not carry k^t, the vector of allocation indicators. However, it is straightforward to obtain smoothed samples of k^t from the full posterior, through an adaptation of the particle smoothing algorithm of Godsill, Doucet, and West (2004).

The particle set $\{\mathcal{Z}_t^{(i)}\}_{i=1}^N$ provides a filtered approximation to the posterior distribution $\mathrm{p}(\mathcal{Z}_t|y^t)$ form which draws from the full posterior distribution of the allocation vector, $\mathrm{p}(k^t|y^t)$, can be obtained through the backwards update equation

$$p(k^t|y^t)=\int \mathrm{p}\left(k^t|\mathcal{Z}_t,y^t\right)d\mathrm{P}(\mathcal{Z}_t|y^t)=\int\prod_{r=1}^{t}\mathrm{p}\left(k_r|\mathcal{Z}_t,y_r\right)d\mathrm{P}(\mathcal{Z}_t|y^t). \tag{10}$$

From (10), we can directly approximate $\mathrm{p}\left(k^t|y^t\right)$ by sampling, for each particle $\mathcal{Z}_t^{(i)}$ and for $r=t,\ldots,1$, k_r with probability $\mathrm{p}\left(k_r=j|\mathcal{Z}_t,y_r\right)\propto\mathrm{p}\left(y_r|k_r=j,\mathcal{Z}_t\right)\mathrm{p}\left(k_r=j|\mathcal{Z}_t\right)$, where j represents each component available in each particle. In the mixture of Poisson example, $\mathrm{p}\left(y_r|k_r=j,\mathcal{Z}_t\right)$ is the density of a Poisson-Gamma for the j component evaluated at y_r and $\mathrm{p}\left(k_r=j|\mathcal{Z}_t\right)$ is equal to n_j/n, which is clearly done independently for each observation. In general, the conditional independence of the mixture models that leads to the factorization in (10) provides an algorithm for posterior allocation that is of order $\mathcal{O}(N)$. This property is not shared by the original proposal of Godsill et al. (2004) where the recursive nature of the dynamic models considered leads to an algorithm of order $\mathcal{O}(N^2)$.

1.3 Marginal Likelihoods

In addition to parameter learning and posterior allocation, PL provides a straightforward mechanism for calculation of the marginal data likelihood associated with a given model specification. This third major inferential tool will be useful in Bayesian model comparison procedures, based either on Bayes factors or posterior model probabilities. A sequential marginal likelihood formulation holds that $\mathrm{p}(y^t) = \prod_{r=1}^{t} \mathrm{p}(y_r|y^{r-1})$. The factors of this product are naturally estimated at each PL step as

$$\mathrm{p}(y_t|y^{t-1}) = \int \mathrm{p}(y_t|\mathcal{Z}_{t-1}) d\mathrm{P}(\mathcal{Z}_{t-1}|y^{t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \mathrm{p}(y_t|\mathcal{Z}_{t-1}^{(i)}). \tag{11}$$

Thus, given each new observation, the marginal likelihood estimate is updated according to $\mathrm{p}(y^t) = \mathrm{p}(y^{t-1}) \sum_{i=1}^{N} \mathrm{p}(y_t|\mathcal{Z}_{t-1}^{(i)})/N$. This approach offers a simple and robust sequential Monte Carlo alternative to the traditionally hard problem of approximating marginal predictive densities via MCMC output (see. e.g., Basu and Chib, 2003; Chib, 1995; Chib and Jeliazkov, 2001; Han and Carlin, 2001).

The following sections will carefully discuss definition of $\mathcal{Z}_t$ and exemplify application of this general PL mixtures strategy in a variety of models.

2 Density Estimation

In this section, we consider density estimation under the class of models characterized by density functions of the form, $f(y; G) = \int \mathrm{k}(y; \theta) dG(\theta)$. There are many possibilities for the prior on G, including the simple finite dimensional models leading to a finite mixture models specification. The most common models, including the very popular Dirichlet process (DP; Ferguson, 1973), are based on the stick-breaking construction for an infinite set of probability weights. Other priors of this type include the beta two-parameter process (Ishwaran and Zarepour, 2000) and kernel stick-breaking processes (Dunson and Park, 2008). Pólya trees (e.g. Paddock et al., 2003) provide an alternative where the distribution is built through a random partitioning of the measurable space. We refer the reader to Walker, Damien, Laud,

and Smith (1999) or Müller and Quintana (2004) for more complete overviews of the major modeling frameworks.

The fundamental requirement in the definition of $\mathcal{Z}_t$ is its ability to allow an "evolution" represented by (5) and analytical evaluation of the predictive in (4). Fortunately, the predictive distribution is central in the development of many nonparametric prior models. Since it is possible to constrain a sequence of distributions $\mathrm{p}(y_{t+1}|y_1,\ldots,y_t)$ so as to obtain exchangeability for the associated sequence y^t (see, e.g., Regazzini, 1998), the predictive probability function can be used to develop probability distributions over y^t through use of de Finetti's representation theorem. Many common nonparametric priors, including the Dirichlet process (Blackwell and MacQueen, 1973) and, more generally, beta-Stacy processes (Walker and Muliere, 1997), Pólya trees (Muliere and Walker, 1997), and species sampling models (Perman et al., 1992) can be characterized in this way. More recently, Lee, Quintana, Müller, and Trippa (2008) propose a general approach to defining predictive probability functions for species sampling models, and argue the flexibility of this model class. Thus, our central use of the predictive probability function fits very naturally within common Bayesian nonparametric modeling frameworks.

In what follows, we are motivated by models where the prior on the mixing distribution is defined via a species sampling model (Pitman, 1995) that guarantees almost surely discrete realizations of G. Making a parallel to (1) and (2), an informal formulation of the collapsed state-space model is

$$\mathbb{E}\left[f(y_{t+1};G)|\mathcal{Z}_t\right] = \int \mathrm{k}(y_{t+1};\theta)d\mathbb{E}\left[G(\theta)\right] \tag{12}$$

$$\mathbb{E}\left[dG(\theta)|\mathcal{Z}_t\right] = \int dG(\theta)d\mathrm{P}\left(dG(\theta)|\mathcal{Z}_t\right). \tag{13}$$

In general, the number of measurable point masses induced by this discrete mixture can be infinite, such that the number of mixture components associated with any observed dataset is random. With t observations allocated to m_t mixture components, (13) can be re-expressed as

$$\mathbb{E}\left[dG(\theta)|\mathcal{Z}_t\right] = p_0 dG_0(\theta) + \sum_{j=1}^{m_t} p_j \mathbb{E}[\delta_{\theta_j^\star}|\mathcal{Z}_t], \tag{14}$$

with $\theta_j^\star$ the parameters for each of the m_t components. This leads to the posterior predictive function

$$\mathrm{p}\left(y_{t+1}|\mathcal{Z}_t\right) = \int \mathrm{k}(y;\theta)\mathbb{E}\left(dG(\theta)|\mathcal{Z}_t\right) \;=\; p_0 \int \mathrm{k}(y_t;\theta)dG_0(\theta) + \sum_{j=1}^{m_t} p_j \int \mathrm{k}(y_t;\theta_j^\star)d\mathrm{P}(\theta_j^\star|\mathcal{Z}_t). \tag{15}$$

The following two sections detail particle learning algorithms for the finite mixture model and Dirichlet process mixture model, both of which lead to predictive probability functions of this type. However, from the above discussion and a few examples described in Section 3, it should be clear the ideas extend to more general nonparametric mixture priors designed around a workable predictive probability function.

2.1 PL for Finite Mixture Models

The density function corresponding to a mixture model with a finite number m of components is written

$$\mathrm{p}(y|\boldsymbol{\theta}^\star, \boldsymbol{p}) = \sum_{j=1}^{m} p_j \mathrm{k}(y;\theta_j^\star), \tag{16}$$

where $\boldsymbol{p} = (p_1, \ldots, p_m)$ such that $\sum_{j=1}^m p_j = 1$ is the mixture component probability vector and $\boldsymbol{\theta}^\star = \{\theta_1^\star, \ldots, \theta_m^\star\}$ is the set of component specific parameters for the density kernel $\mathrm{k}(\cdot;\theta)$. This is clearly a special case of (12) where the random mixing distribution degenerates at m distinct atoms. Equivalently, this implies that the density representation can be written as $\mathbb{E}\left(dG(\theta)|\mathcal{Z}_t\right) = \sum_{j=1}^m p_j \delta_{\theta_t^\star}$, so that the quantity of interest is the joint posterior distribution $\mathrm{p}(\boldsymbol{\theta}^\star, \boldsymbol{p}|y)$. The model is completed with independent prior distributions $\pi(\boldsymbol{\theta}^\star;\psi_\theta)$ and $\pi(\boldsymbol{p};\psi_p)$. The hyper-parameters, $\boldsymbol{\psi} = \{\psi_\theta, \psi_p\}$, may also be treated as random and assigned prior distributions. In this case, $\boldsymbol{\psi}$ is included in the essential state vector and it is resampled off-line (after the propagation step) from its full conditional. In particular, posterior inference for random prior parameters is based upon the posterior distributions $\mathrm{p}(\psi_\theta|\boldsymbol{\theta}^\star)$ and $\mathrm{p}(\psi_p|\boldsymbol{p})$, conditionally independent of the data y^t.

Given a vector of t observations, $y^t = (y_1, \ldots, y_t)$, that are assumed to have been sampled i.i.d. with density as in (16), the standard approach to inference is to break the mixture through the introduction of a latent allocation vector $k^t = (k_1, \ldots, k_t)$ such that $\mathrm{p}(y_t|k_t, \boldsymbol{\theta}^\star) =$

$\mathrm{k}(y_t; \theta^\star_{k_t})$. Introduction of the latent allocation vector leads to conditionally independent posterior distributions $\mathrm{p}(\boldsymbol{\theta}^\star|k^t, y^t)$ and $\mathrm{p}(\boldsymbol{p}|k^t)$. When $\pi(\boldsymbol{\theta}^\star) = \prod_{j=1}^m \pi(\theta^\star_j)$, as is standard, the posterior for mixture component parameters separates into $\mathrm{p}(\boldsymbol{\theta}^\star|k^t, y^t) = \prod_{j=1}^m \mathrm{p}(\theta^\star_j|\{y_r : k_r = j\})$. In many settings, conditionally conjugate priors are chosen such that sampling from these distributions is straightforward. Regardless, posterior sampling will rely upon this conditional independence structure and it is clear that basic inference for mixture models lies in posterior draws for k^t. All else proceeds conditional upon these sampled latent allocations.

The posterior information that is available given k^t can be summarized by the number of observations allocated to each component, $\mathbf{n}_t = (n_{t,1}, \ldots n_{t,m})$, and the conditional sufficient statistics for the mixture component parameters, $\mathbf{s}_t = (s_{t,1}, \ldots s_{t,m})$, such that $n_{t,j}$ and $s_{t,j}$ are sufficient for $\theta^\star_j$ given y^t. We can therefore define $\mathcal{Z}_t = (k_t, \mathbf{s}_t, \mathbf{n}_t)$ as our state vector. Posterior inference for each "time" $t+1$ thus leads to a sequence of updated posteriors,

$$\mathrm{p}(k_{t+1}, \mathbf{s}_{t+1}, \mathbf{n}_{t+1}|y^{t+1}) = \int \mathrm{p}(k_{t+1}, \mathbf{s}_{t+1}, \mathbf{n}_{t+1}|\mathbf{s}_t, \mathbf{n}_t, y_{t+1}) d\mathrm{P}(\mathbf{s}_t, \mathbf{n}_t|y^{t+1}). \tag{17}$$

After going through all samples ($t = T$), completed filtering provides an estimate of the posterior for $(\mathbf{s}_T, \mathbf{n}_T)$, and the conditional independence structure of mixture models leads to straightforward Rao-Blackwellized sampling from $\mathrm{p}(\boldsymbol{\theta}^\star, \boldsymbol{p}|y^T) = \int \mathrm{p}(\boldsymbol{\theta}^\star, \boldsymbol{p}|\mathbf{s}_T, \mathbf{n}_T) d\mathrm{P}(\mathbf{s}_T, \mathbf{n}_T|y^T)$.

It is important to notice that $\mathrm{p}(k_{t+1}, \mathbf{s}_{t+1}, \mathbf{n}_{t+1}|\mathbf{s}_t, \mathbf{n}_t, y_{t+1}) = \mathrm{p}(k_{t+1}|\mathbf{s}_t, \mathbf{n}_t, y_{t+1})$, by virtue of deterministic mappings

$$s_{t+1,k_{t+1}} = \mathcal{S}(s_{t,k_{t+1}}, y_{t+1}) \quad \text{and} \quad n_{t+1,k_{t+1}} = n_{t,k_{t+1}} + 1, \tag{18}$$

where information for components $j \neq k_{t+1}$ remains the same and $\mathcal{S}$ is determined by the mixture kernel. Thus the target integrand in (17) is the product of a posterior full conditional for k_{t+1} and the predictive term $\mathrm{p}(\mathbf{s}_t, \mathbf{n}_t|y^{t+1}) \propto \mathrm{p}(y_{t+1}|\mathbf{s}_t, \mathbf{n}_t)\mathrm{p}(\mathbf{s}_t, \mathbf{n}_t|y^t)$. It should be clear that this is exactly the representation in (4) and (5) so that we can develop a version of general algorithm presented in Section 2.

Before moving on, we note that implementation of PL always relies on availability of the predictive and "propagation" distributions. In the finite mixture set-up these are supplied by $\mathrm{p}\left(y_{t+1}|\mathbf{s}_t, \mathbf{n}_t\right)$, $\mathrm{p}\left(k_{t+1}|\mathbf{s}_t, \mathbf{n}_t, y_{t+1}\right)$ and the deterministic recursions in (18). The latter can

always be obtained as k_{t+1} is a discrete variable, and the former will be available directly whenever $\pi(\theta_j^\star)$ is conjugate for $k(y;\theta_j^\star)$ and $\pi(\boldsymbol{p})$ is conjugate for multinomial data. However, despite our focus on sufficient statistics, the uncertainty update in (17) is valid even in models without full conditional conjugacy. In such situations, $s_{t,j}$ may just be the data subset $\{y_r : k_r = j\}$. Regardless, it will be possible to obtain a version of (17) through the introduction of a set of auxiliary variables in $\mathcal{Z}_t$, possibly a function of $\boldsymbol{\theta}^\star$ or $\boldsymbol{p}$, sampled conditional on $\mathbf{s}_t$ and $\mathbf{n}_t$. In fact, one can think of the members of $\mathcal{Z}_t$ as general information states, containing both conditional sufficient statistics and any auxiliary variables; that is, whatever is required to evaluate the posterior predictive and sample from the full conditional. We exemplify this notion at the end of DP mixture of multivariate normals example (Section 4.2) where we discuss a situation in which hyperparameters are learned and the concentration parameter α is unknown.

Given a set of particles $\left\{\mathbf{n}_t^{(i)}, \mathbf{s}_t^{(i)}\right\}_{i=1}^N$ approximating $\mathrm{p}\left(\mathbf{n}_t, \mathbf{s}_t | y^t\right)$ and a new observation y_{t+1}, the PL algorithm for finite mixture models updates the approximation to $\mathrm{p}\left(\mathbf{n}_{t+1}, \mathbf{s}_{t+1} | y^{t+1}\right)$ using the following resample/propagation rule.

Algorithm 1: PL for finite mixture models

1. **Resample**: Generate an index $\boldsymbol{\zeta} \sim \mathrm{MN}(\boldsymbol{\omega}, N)$ where

$$\omega(i) = \frac{\mathrm{p}\left(y_{t+1} | (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)}{\sum_{i=1}^N \mathrm{p}\left(y_{t+1} | (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)}$$

2. **Propagate**:

$$\begin{aligned} k_{t+1} &\sim \mathrm{p}\left(k_{t+1} | (\mathbf{s}_t, \mathbf{n}_t)^{\zeta(i)}, y_{t+1}\right) \\ \mathbf{s}_{t+1} &= \mathcal{S}\left(\mathbf{s}_t^{\zeta(i)}, k_{t+1}, y_{t+1}\right) \\ n_{t+1,k_{t+1}} &= n_{t,k_{t+1}}^{\zeta(i)} + 1, \quad n_{t+1,j} = n_{t,j}^{\zeta(i)} \quad \text{for} \quad j \neq k_{t+1} \end{aligned}$$

3. **Learn**:

$$\mathrm{p}(\boldsymbol{p}, \boldsymbol{\theta}^\star | y^t) = \frac{1}{N} \sum_{i=1}^N \mathrm{p}\left(\boldsymbol{p}, \boldsymbol{\theta}^\star | (\mathbf{s}_t, \mathbf{n}_t)^{(i)}\right)$$

Once again, if $\mathrm{p}(k^T|y^T)$ is of interest, samples can be directly obtained from the backwards uncertainty update equation

$$\begin{aligned} p(k^T|y^T) &= \int \mathrm{p}\left(k^T|\mathbf{s}_T, \mathbf{n}_T, y^T\right) \mathrm{p}\left(\mathbf{s}_T, \mathbf{n}_T|y^T\right) d(\mathbf{s}_T, \mathbf{n}_T) \\ &= \int \prod_{t=1}^{T} \mathrm{p}\left(k_t|\mathbf{s}_T, \mathbf{n}_T, y_t\right) \mathrm{p}\left(\mathbf{s}_T, \mathbf{n}_T|y^T\right) d(\mathbf{s}_T, \mathbf{n}_T). \end{aligned} \tag{19}$$

From this, we can directly approximate $\mathrm{p}\left(k^T|y^T\right)$ by sampling, for each particle $(\mathbf{s}_T, \mathbf{n}_T)^{(i)}$ and for $t = 1, \ldots, T$, k_t with probability $\mathrm{p}\left(k_t = j|\mathbf{s}_T, \mathbf{n}_T, y_t\right)$ proportional to $\mathrm{p}\left(y_t|k_t = j, \mathbf{s}_T\right) \mathrm{p}\left(k_t = j|\mathbf{n}_T\right)$.

2.2 PL for Nonparametric Mixture Models

Discrete nonparametric mixture models have, since the early work of Ferguson (1974) and Antoniak (1974), emerged as a dominant modeling tool for Bayesian nonparametric density estimation (see also Ferguson, 1983; Lo, 1984). Unlike in finite mixture models, the number of unique mixture components is random. For this reason, both m_t and $\boldsymbol{\theta}_t^\star$ now depend upon the "time" t. Analogously to the finite setting, the posterior information that is available conditional on k^t (and implicitly m_t) can be summarized by $\mathbf{n}_t = (n_{t,1}, \ldots, n_{t,m_t})$, the number of observations allocated to each unique component, and $\mathbf{s}_t = (s_{t,1}, \ldots, s_{t,m_t})$, the conditional sufficient statistics for the component parameters. The state vector to be tracked by PL can then be defined as $\mathcal{Z}_t = (k_t, m_t, \mathbf{s}_t, \mathbf{n}_t)$.

The standard approach to inference is to use a collapsed Gibbs sampler that cycles through draws of $\mathrm{p}(k_t|\mathbf{n}_T^{(-t)}, \mathbf{s}_T^{(-t)}, m_T^{(-t)}, y_t)$, where $\mathbf{n}_T^{(-t)}$ and $\mathbf{s}_T^{(-t)}$ denote each respective set with the t^{th} element removed, for $t = 1, \ldots, T$, and $m_T^{(-t)}$ is the implied number of distinct components. In the case of DP mixture models, this algorithm has been detailed in Escobar and West (1995). Extension to the case of conditionally nonconjugate kernel models is described by Bush and MacEachern (1996), and Neal (2000) provides an overview of more state-of-the-art versions of the algorithm. Furthermore, Ishwaran and James (2001) describe sampling for truncated G approximations that can be utilized whenever it is not possible to marginalize over unallocated mixture components (note that these approximations can be fit with the finite mixture PL approach of Section 2.1).

Our framework extends to infinite mixture models in a very straightforward manner, and particle learning for nonparametric mixture models proceeds through the two familiar steps:

$$\text{Resample } (\mathbf{s}_t, \mathbf{n}_t, m_t) \propto \mathrm{p}(y_{t+1}|\mathbf{s}_t, \mathbf{n}_t, m_t) \quad \rightarrow \quad \text{Propagate } k_{t+1} \sim \mathrm{p}(k_{t+1}|\mathbf{s}_t, \mathbf{n}_t, m_t, y_{t+1}). \tag{20}$$

Indeed, PL will apply to any nonparametric mixture where the two equations in (20) are available either analytically or approximately. The filtered posterior for $(\mathbf{s}_T, \mathbf{n}_T, m_T)$ can be used for inference via the posterior predictive density $\mathrm{p}(y|\mathbf{s}_T, \mathbf{n}_T, m_T)$, which is a Rao-Blackwellized version of $\mathbb{E}[f(y;G)|y^T]$ for many nonparametric priors (including the DP). Alternatively, since $\mathrm{p}(G|y^T) = \int \mathrm{p}(G|\mathbf{s}_T, \mathbf{n}_T, m_T)\ d\mathrm{P}(\mathbf{s}_T, \mathbf{n}_T, m_T|y^T)$, the filtered posterior provides a basis for inference about the full random mixing distribution.

To make the presentation more concrete we concentrate on a PL framework for Dirichlet Process mixtures, the most commonly used nonparametric prior for random mixture models. The DP characterizes a prior over probability distributions and is most intuitively represented through its constructive definition (Perman et al., 1992): a random distribution G generated from $\mathrm{DP}(\alpha, G_0(\psi))$ is almost surely of the form

$$dG(\cdot) = \sum_{l=1}^{\infty} p_l\, \delta_{\vartheta_l}(\cdot) \text{ with } \vartheta_l \overset{iid}{\sim} G_0(\vartheta_l;\psi),\ p_l = (1-\sum_{j=1}^{l-1} p_j)v_l, \text{ and } v_l \overset{iid}{\sim} \mathrm{beta}(1,\alpha) \text{ for } l = 1, 2, \ldots \tag{21}$$

where $G_0(\vartheta;\psi)$ is the centering distribution function, parametrized by ψ, and the sequences $\{\vartheta_l, l = 1, 2, \ldots\}$ and $\{v_k : l = 1, 2, ...\}$ are independent. The discreteness of DP realizations is explicit in this definition.

The DP mixture model for y^t is then $f(y_r;G) = \int \mathrm{k}(y_r;\theta)dG(\theta)$ for $r = 1, \ldots, t$, where $G \sim \mathrm{DP}(\alpha, G_0)$. Alternatively, in terms of latent variables, the hierarchical model is that for $r = 1, \ldots, t$, $y_r \overset{ind}{\sim} k(y_r;\theta_r)$, $\theta_r \overset{iid}{\sim} G$ and $G \sim \mathrm{DP}(\alpha, G_0)$. Recall from definitions above that $\boldsymbol{\theta}_t^\star = \{\theta_1^\star, \ldots, \theta_{m_t}^\star\}$ is the set of m_t distinct components in θ^t, k^t is the associated latent allocation such that $\theta_t = \theta_{k_t}^\star$, $\mathbf{n}_t = (n_{t,1}, \ldots, n_{t,m_t})$ is the number of observations allocated to each unique component, and $\mathbf{s}_t = (s_{t,1}, \ldots, s_{t,m_t})$ is the set of conditional sufficient statistics for each $\theta_j^\star$.

Two properties of the DP are particularly important for sequential inference. First, the

DP is a conditionally conjugate prior: given θ^t (or, equivalently, $\boldsymbol{\theta}^\star_t$ and $\mathbf{n}_t$), the posterior distribution for G is characterized as a $\mathrm{DP}(\alpha + t, G_0^t)$ where,

$$dG_0^t(\theta; \boldsymbol{\theta}^\star_t, \mathbf{n}_t) = \frac{\alpha}{\alpha + t} dG_0(\theta) + \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha + t} \delta_{[\theta = \theta^\star_j]}. \tag{22}$$

Second, this Pólya urn density dG_0^t is also $\mathbb{E}[\ dG|\theta^t\ ] = \int dG(\theta) d\mathrm{P}(G|\boldsymbol{\theta}^\star_t, \mathbf{n}_t)$, and provides a finite predictive probability function for our mixture model: $\mathrm{p}(y_{t+1}|\theta^t) = \int \mathrm{k}(y_{t+1};\theta) dG_0^t(\theta)$.

As always, we focus on sequential inference for conditional sufficient statistics. Similar to the version for finite mixtures, the uncertainty updating equation is $\mathrm{p}(\mathbf{s}_{t+1}, \mathbf{n}_{t+1}|y^{t+1}) \propto \int \mathrm{p}(\mathbf{s}_{t+1}, \mathbf{n}_{t+1}|\mathbf{s}_t,\ \mathbf{n}_t,\ m_t,\ y_{t+1})\ \mathrm{p}(y_{t+1}|\mathbf{s}_t, \mathbf{n}_t, m_t) d\mathrm{P}(\mathbf{s}_t, \mathbf{n}_t, m_t|y^t)$, with posterior predictive density

$$\begin{aligned} \mathrm{p}(y_{t+1}|\mathbf{n}_t, \mathbf{s}_t) &= \iiint \mathrm{k}(y_{t+1};\theta) dG(\theta) d\mathrm{P}(G|\boldsymbol{\theta}^\star_t, \mathbf{n}_t) d\mathrm{P}(\boldsymbol{\theta}^\star_t|\mathbf{n}_t, \mathbf{s}_t) \\ &= \int \mathrm{k}(y_{t+1};\theta) \left[\int dG_0^t(\theta; \boldsymbol{\theta}^\star_t, \mathbf{n}_t) d\mathrm{P}(\boldsymbol{\theta}^\star_t|\mathbf{n}_t, \mathbf{s}_t) \right]. \end{aligned} \tag{23}$$

By virtue of a deterministic mapping, $\mathrm{p}(\mathbf{s}_{t+1}, \mathbf{n}_{t+1}, m_{t+1}|\mathbf{s}_t, \mathbf{n}_t, m_t, y_{t+1})$ is just $\mathrm{p}(k_{t+1}|\mathbf{s}_t,\ \mathbf{n}_t,\ m_t,\ y_{t+1})$. From the argument in (23), $\mathrm{p}(k_{t+1} = j|\mathbf{s}_t, \mathbf{n}_t, m_t, y_{t+1})$ is proportional to

$$\begin{aligned} n_{t,j} \int \mathrm{k}(y_{t+1};\theta^\star_j) d\Pr(\theta^\star_j|s_{t,j}, n_{t,j}) \quad &\text{for} \quad j = 1, \ldots, m_t \\ \text{and} \ \ \alpha \int \mathrm{k}(y_{t+1};\theta) dG_0(\theta) \quad &\text{if} \quad j = m_t + 1. \end{aligned} \tag{24}$$

With (23) and (24) defined, a particle learning approach is straightforward. Assuming that current particles $\{(\mathbf{n}_t, \mathbf{s}_t, m_t)^{(i)}\}_{i=1}^N$ approximate the posterior $\mathrm{p}(\mathbf{s}_t, \mathbf{n}_t, m_t|y^t)$, the algorithm for updating posterior uncertainty is summarized as Algorithm 2, below.

The resample and propagate steps should look familiar, as they are a straightforward extension of the steps in Algorithm 1. The third step here, estimation, presents only one possible type of inference; Section 4.3 describes and illustrates inference about G conditional on a filtered set of particles. However, in many situations, density estimation is the primary goal of DP mixture modeling, and step 3 shows that the filtered particle set immediately provides a posterior sample of density estimates through the predictive probability function.

Note that this is just a Rao-Blackwellized version of the standard Pólya urn mixture that serves as a density estimator: $\mathrm{p}\left(\mathbb{E}[f(y;G)]|y^t\right) = \int \mathrm{p}\left(\mathbb{E}[f(y;G)]|\mathbf{s}_t, \mathbf{n}_t, m_t\right) d\mathrm{P}(\mathbf{s}_t, \mathbf{n}_t, m_t|y^t)$, and $\mathrm{p}\left(\mathbb{E}[f(y;G)]|\mathbf{s}_t, \mathbf{n}_t, m_t\right) = \int \mathrm{p}(y|\boldsymbol{\theta}_t^\star, \mathbf{n}_t)\ d\mathrm{P}(\boldsymbol{\theta}_t^\star|\mathbf{s}_t, \mathbf{n}_t, m_t)$. Finally, note that if either α or ψ are assigned hyperpriors, these can be included in $\mathcal{Z}_t$ and sampled off-line for each particle conditional on $(\mathbf{n}_t, \mathbf{s}_t, m_t)^{(i)}$ at each iteration. This is of particular importance in the understanding of the generality of PL and it is described clearly in Section 4.2.

Algorithm 2: PL for DP mixture models

1. **Resample**: Generate an index $\boldsymbol{\zeta} \sim \mathrm{MN}(\boldsymbol{\omega}, N)$ where

$$\omega(i) = \frac{\mathrm{p}\left(y_{t+1}|(\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)}{\sum_{i=1}^{N} \mathrm{p}\left(y_{t+1}|(\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)}$$

2. **Propagate**:

$$k_{t+1} \sim \mathrm{p}\left(k_{t+1}|(\mathbf{s}_t, \mathbf{n}_t, m_t)^{\zeta(i)}, y_{t+1}\right),$$

$\mathbf{s}_{t+1} = \mathcal{S}\left(\mathbf{s}_t, k_{t+1}, y_{t+1}\right)$. For $j \neq k_{t+1}$, $n_{t+1,j} = n_{t,j}$.

If $k_{t+1} \leq m_t$, $n_{t+1,k_t} = n_{t,k_t} + 1$ and $m_{t+1} = m_t$.

Otherwise, $m_{t+1} = m_t + 1$ and $n_{t,m_{t+1}} = 1$.

3. **Estimation**:

$$\mathrm{p}\left(\mathbb{E}[f(y;G)]|y^t\right) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{p}\left(y|(\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)}\right)$$

3 Other Nonparametric Models

3.1 Latent Features Models via the Indian Buffet Process

Latent feature models attempt to describe an object y_t using a set of latent features. For data $\mathbf{Y}^T = \{y_1', y_2', \ldots, y_T'\}$, where each y_t is a p-dimensional column vector, these models assume that $\mathbf{Y}^T \approx \mathbf{Z}^T\mathbf{B}$ where $\mathbf{Z}^T$ is a $T \times k$ matrix indicating which latent features are present in each object and $\mathbf{B}$ is a $k \times p$ matrix defining how these features are expressed. Typically, k is much smaller than p so that the goal of finding a lower dimensional representation for $\mathbf{Y}^T$ is achieved. This is, of course, a traditional framework in statistical data analysis as it essentially a factor model.

This is a very popular tool in machine learning where there exists a great focus on models derived from processes defined over infinite binary random matrices where k does not have to be specified a priori and instead can be learned from the data. This is accomplished with the use of nonparametric Bayesian tools that assume $k = \infty$ a priori but guarantees it to be finite and smaller than T *a posteriori*. Perhaps the most popular approach in this class of models is the Indian Buffet Process put forward by Griffiths and Ghahramani (2006). Similarly to the Dirichlet process, the IBP can be formulated through a sequential generative scheme which in turn makes it suitable to be solved with PL. In a nutshell, imagine an Indian restaurant where costumers arrive one by one and head to the buffet. The first costumer selects a number of $\text{Po}(\alpha)$ of dishes. The t^{th} costumer will select dishes proportionally to its popularity so far with probability $\frac{n_{t,j}}{t}$ where n_j is the number of costumers that have sampled the j^{th} dish. Then, the same costumer, chooses a number $\text{Po}(\alpha/t)$ of new dishes. This is can be easily re-interpreted a Pólya urn scheme where dishes represent latent features and costumers represent observations (or objects).

To make this discussion more concrete, we focus on the infinite linear-Gaussian matrix factorization model of Griffiths and Ghahramani (2006):

$$\mathbf{Y}^T = \mathbf{Z}^T\mathbf{B} + \mathbf{E} \tag{25}$$

where $\mathbf{E}$ is a $T \times p$ matrix of independent normal errors $e_{ij} \sim \text{N}(0, \sigma^2)$. Assume further that

the prior for each element of $\mathbf{B}$ is $\mathrm{N}(0, \sigma^2\sigma_b^2)$ and $\mathbf{Z}^T$ is a $T \times k$ matrix of binary features. Without loss of generality, assume knowledge of both σ^2 and σ_b^2. This is simplified version of models used to analyze image data where each object is a image containing p pixels that are to be modeled via a smaller number of latent features.

By stating that $\mathbf{Z}^T$ arises from an IBP (with parameter α) we can simply recast (25) as a state-space model where each row of $\mathbf{Z}^T$ is a state and the transition $\mathrm{p}(z_{t+1}|\mathbf{Z}^T)$ follows from the IBP generative process as described above.

Define the essential state vector $\mathcal{Z}_t = (m_t, \mathbf{s}_t, \mathbf{n}_t)$ where m_t is the number of current latent features, $\mathbf{n}_t$ is the number of objects allocated to each of the currently available latent features and $\mathbf{s}_t$ is the set of conditional sufficient statistics for $\mathbf{B}$. Before evaluating the posterior predictive for y_{t+1} it is necessary to propose a new number m_* of potential new features (dishes) per particle. From the definition of the IBP it follows that for each particle i, $m_*^{(i)} \sim Po(\frac{\alpha}{t})$ in turn defining a $m_t + m_*$ latent feature model. The posterior predictive necessary to re-sample the particles can be evaluated via

$$\mathrm{p}\left(y_{t+1}|\mathcal{Z}_t, m_*\right) \propto \sum_{z_{t+1}\in\mathcal{C}} \mathrm{p}\left(y_{t+1}|z_{t+1}, \mathbf{s}_t\right) \mathrm{p}\left(z_{t+1}|\mathcal{Z}_t\right) \tag{26}$$

where the set $\mathcal{C}$ is defined by all 2^{m_t} possible configurations of z_{t+1} where the final m_* elements are fixed at one. Notice that the terms inside of the summation sign are both available in close form. First, with the conditionally conjugate structure of the model, it is possible to evaluate the likelihood marginally on $\mathbf{B}$. Further, the prior for z_{t+1} is simply a product of independent probabilities each defined proportionally to $n_{t,j}/t$.

Propagating $\mathcal{Z}_{t+1}$ forward is also very simple. First, draw for each particle the configuration for z_{t+1} from

$$\mathrm{p}\left(z_{t+1}|\mathcal{Z}_t, y_{t+1}\right) \propto \mathrm{p}\left(y_{t+1}|z_{t+1}, \mathbf{s}_t, \right) \mathrm{p}\left(z_{t+1}|\mathcal{Z}_t\right)$$

and update $n_{t+1,j} = n_{t,j} + 1$ if $z_{t+1,j} = 1$, $\mathbf{s}_t = \mathcal{S}\left(\mathbf{s}_t, y_{t+1}, z_{t+1}\right)$, and $m_{t+1} = m_t + m_*$. Once again, $\mathcal{S}(\cdot)$ is a deterministic function based on standard Gaussian updates.

3.2 Dependent Nonparametric Mixture Models

The PL mixtures approach can also be adapted to simulation for dependent DP mixtures, as introduced by MacEachern (2000). In the setting that you have multiple mixture densities, $f_s(y) = \int \mathrm{k}(y;\boldsymbol{\theta})dG_s(\boldsymbol{\theta})$ for some index $s = 1,\ldots,S$ (e.g., discrete time or spatial locations), the dependent DP mixture model holds that each dG_s is realized $\sum_{l=1}^{\infty} p_{ls}\delta_{\vartheta_{ls}}(\theta)$ – that is, as in (21) but now with possible dependence across s-values for the kernel parameters or the stick-breaking weights. In the most common formulation of this framework, stick-breaking weights are constant in s and the combined kernel parameters $\boldsymbol{\vartheta}_l = \{\vartheta_{l1}\ldots\vartheta_{lS}\}$ are drawn jointly from a $\dim(\theta)\times S$ dimension centering distribution $G_0(\boldsymbol{\vartheta})$. In posterior simulation for such 'single-θ' models, one is able treat everything as a standard DP mixture model for density estimation over the expanded $\dim(y)\times S$ observation space; refer to Gelfand et al. (2005) for an example of this approach. Hence, the techniques developed in Section 2.2 apply directly.

In addition, a number of different approaches have recently been proposed for the construction of dependent nonparametric mixture models with correlated stick-breaking weights (see, e.g., Griffin and Steel, 2006; Rodriguez and Dunson, 2009). Although these models tend to be more difficult to fit through MCMC than the single-θ type schemes, it is often possible to develop fairly straightforward PL simulation strategies. One successful approach is outlined in (Taddy, 2010), where PL was used in mean-inference for discrete-time autoregressive stick-breaking mixtures. In this framework, a series of correlated mixing distributions, G_t for $t = 1,\ldots,T$, are marginally distributed as a DP but with stick-breaking proportions that are drawn from an autoregressive time series of beta random variables. In detail, the model is as in (21) except that each series of stick-breaking weights, $\mathbf{v}_l = [v_{l1}\ldots v_{lT}]$ is modeled as a Beta Autoregressive Process (introduced by McKenzie, 1985). Section 3.2 of Taddy (2010) details the PL algorithm for this model. In contrast with the algorithms of 2.2, it is not possible to integrate over all of the stick-breaking weights, and a finite number of these weights must be included in the particle set. Interestingly, through some careful steps to avoid particle degeneracy, the author shows that it is possible to use PL to sample and make inference about the mixture weights themselves (e.g., see Figure 7 of the Taddy paper).

To further illustrate use of PL with these types of models, we will sketch the algorithm for a probit stick-breaking framework proposed by Rodriguez and Dunson (2009). Here, mixture

weights are built through probit transformations of underlying latent variables, allowing for the introduction of spatial and temporal dependence structure. In particular, the discrete-time model is defined as, for $t = 1, \ldots, T$, $y_{t+1} \sim \int \mathrm{k}(y_{t+1}|\boldsymbol{\theta}) dG_{t+1}(\boldsymbol{\theta})$ and $\mathbb{E}\left[dG_{t+1}(\theta)|\mathcal{Z}_{t+1}\right] = w_{0,t+1} dG_0(\theta) + \sum_{j=1}^{m_t} w_{j,t+1} \delta_{\theta_t^\star}$ where $w_{j,t+1} = \Phi(\alpha_{j,t+1}) \prod_{r<j} (1 - \Phi(\alpha_{r,t+1}))$, $\alpha_{j,t+1} = \mathbf{A}_{t+1}\boldsymbol{\eta}_{t+1}$, $\boldsymbol{\eta}_{j,t+1} = \mathbf{B}_{t+1}\boldsymbol{\eta}_{j,t} + \boldsymbol{\nu}_{j,t+1}$, and $\boldsymbol{\nu}_{j,t} \sim \mathrm{N}(0, \mathbf{W}_t)$. We assume knowledge of the parameters defining the dynamic linear model (West and Harrison, 1997) $\{\mathbf{A}_t, \mathbf{B}_t, \mathbf{W}_t\}$ for all t. By appropriately defining these quantities one can embed a variety of different behaviors in the evolution of the non-parametric distribution, including trends, periodicity, autoregression, etc.

To illustrate the use of PL we work with dependent probit-stick breaking priors with latent Markov structure. These are models for distributions that evolve in discrete time. The model is defined as, for $t = 1, \ldots, T$, $y_{t+1} \sim \int \mathrm{k}(y_{t+1}|\boldsymbol{\theta}) dG_{t+1}(\boldsymbol{\theta})$ and $\mathbb{E}\left[dG_{t+1}(\theta)|\mathcal{Z}_{t+1}\right] = w_{0,t+1} dG_0(\theta) + \sum_{j=1}^{m_t} w_{j,t+1} \delta_{\theta_t^\star}$ where $w_{j,t+1} = \Phi(\alpha_{j,t+1}) \prod_{r<j} (1 - \Phi(\alpha_{r,t+1}))$, $\alpha_{j,t+1} = \mathbf{A}_{t+1}\boldsymbol{\eta}_{t+1}$, $\boldsymbol{\eta}_{j,t+1} = \mathbf{B}_{t+1}\boldsymbol{\eta}_{j,t} + \boldsymbol{\nu}_{j,t+1}$, and $\boldsymbol{\nu}_{j,t} \sim \mathrm{N}(0, \mathbf{W}_t)$. We assume knowledge of the parameters defining the dynamic linear model (West and Harrison, 1997) $\{\mathbf{A}_t, \mathbf{B}_t, \mathbf{W}_t\}$ for all t. By appropriately defining these quantities one can embed a variety of different behaviors in the evolution of the non-parametric distribution, including trends, periodicity, autoregression, etc.

What makes the above representation very attractive is the fact that we can effectively work with the probit link defining the stick-breaking weights by the commonly used data augmentation trick where

$$w_{j,t+1}^* = \mathbb{1}(z_{j,t+1} < 0) \prod_{r<j} \mathbb{1}(z_{r,t+1} > 0) \tag{27}$$

such that

$$z_{j,t+1} = \mathbf{A}_{t+1}\boldsymbol{\eta}_{j,t+1} + \epsilon_{j,t+1} \tag{28}$$

$$\boldsymbol{\eta}_{j,t+1} = \mathbf{B}_{t+1}\boldsymbol{\eta}_{j,t} + \boldsymbol{\nu}_{j,t+1} \tag{29}$$

with $\epsilon_{j,t+1} \sim \mathrm{N}(0, 1)$. This facilitates the posterior sampling of such models and it is also useful for sequential inferences via PL.

Define the essential state vector $\mathcal{Z}_t = (\mathbf{H}_t, m_t, \mathbf{S}_t)$ where $\mathbf{H}_t$ is the collection of m_t current

instances of $\boldsymbol{\eta}_t$ and, as usual, $\mathbf{S}_t$ is the set of conditional sufficient statistics for $\boldsymbol{\theta}$ (here again, we assume conjugate kernels). The predictive distribution can be defined as

$$\mathrm{p}\left(y_{t+1}|\mathcal{Z}_t\right) = \sum_{k_{t+1}=1}^{m_t} \mathrm{p}\left(y_{t+1}|k_{t+1}, \mathbf{S}_t\right) \mathrm{p}\left(k_{t+1}|\mathcal{Z}_t\right) + \mathrm{p}\left(k_{t+1} = m_t + 1\right) \int \mathrm{p}\left(y_{t+1}|\boldsymbol{\theta}_t\right) dG_0(\boldsymbol{\theta})$$

where k_{t+1} is an indicator or the mixture component and

$$\mathrm{p}\left(k_{t+1} = l|\mathcal{Z}_t\right) = \Phi\left(-\frac{\mathbf{A}_{t+1}\mathbf{B}_{t+1}\boldsymbol{\eta}_{l,t}}{\mathbf{A}'_{t+1}\mathbf{W}_{t+1}\mathbf{A}_{t+1}+1}\right) \prod_{r<l}\left[1 - \Phi\left(\frac{-\mathbf{A}_{t+1}\mathbf{B}_{t+1}\boldsymbol{\eta}_{r,t}}{\mathbf{A}'_{t+1}\mathbf{W}_{t+1}\mathbf{A}_{t+1}+1}\right)\right].$$

Propagating $\mathcal{Z}_t$ forward starts by sampling the allocation k_{t+1} with probability

$$\mathrm{p}\left(k_{t+1} = l|\mathcal{Z}_t, y_{t+1}\right) \propto \begin{cases} \mathrm{p}\left(y_{t+1}|k_{t+1} = l, \mathbf{S}_t\right) \mathrm{p}\left(k_{t+1} = l|\mathcal{Z}_t\right) & \text{for} \quad l = 1, \ldots, m_t \\ \mathrm{p}\left(k_{t+1} = m_t + 1\right) \int \mathrm{p}\left(y_{t+1}|\boldsymbol{\theta}_t\right) dG_0(\boldsymbol{\theta}) \end{cases}.$$

Next, with k_{t+1} in hand, both m_{t+1} and $\mathbf{S}_{t+1}$ are deterministically updated. The final propagation step involves sampling $\mathbf{H}_{t+1}$. Here's where the data augmentation step described above becomes relevant. By sampling $z_{j,t+1}$ for all j, (28) and (29) turn into a simple dynamic linear model, with straightforward updates for the posterior $\mathrm{p}\left(\boldsymbol{\eta}_{j,t+1}|z_{j,t+1}\right)$. From (27) we see that $k_{t+1} = j$ leads to $z_{j,t+1} > 0$ and $z_{r,t+1} < 0$ for all $r < j$ and therefore the data augmentation variables can be generated from

$$\mathrm{p}\left(z_{l,t+1}|k_{t+1}\right) = \begin{cases} \mathrm{N}\left(\mathbf{A}_{t+1}\mathbf{B}_{t+1}\boldsymbol{\eta}_t, \mathbf{A}'_{t+1}\mathbf{W}_{t+1}\mathbf{A}_{t+1}+1\right) \mathbb{1}(z_{l,t+1} > 0) & \text{for} \quad l = k_{t+1} \\ \mathrm{N}\left(\mathbf{A}_{t+1}\mathbf{B}_{t+1}\boldsymbol{\eta}_t, \mathbf{A}'_{t+1}\mathbf{W}_{t+1}\mathbf{A}_{t+1}+1\right) \mathbb{1}(z_{l,t+1} < 0) & \text{for} \quad l < k_{t+1} \end{cases}.$$

4 Examples

4.1 Finite Mixture of Poisson Densities

We start with an application of PL in a Poisson finite mixture model as described in Example 1 of Section 1. Figure 1 shows the simulated data and inferred predictive densities obtained by PL, alongside Bayes factors for the number of mixture components, calculated through the marginal likelihood estimation procedure of Section 1.1. The central panel displays the true

predictive distribution as well as inferred versions of it with m set to 2, 3 and 4. This shows that predictive estimates from PL are very similar to the ones obtain by traditional MCMC methods and, in this particular case, also very close to the truth. In addition, the right hand side of Figure 1 shows a simulation study where Bayes factors for choosing between $m = 3$ or $m = 4$ are repeatedly computed via different methods. The data is always the same and the variation arises from the Monte Carlo variation incurred by each algorithm. In the graph, "MCMC" stands for fitting the model using a traditional data augmentation Markov chain Monte Carlo scheme followed by the use of the methods proposed in Basu and Chib (2003). It is important to highlight that this methodology requires the use of a sequential importance sampling and, in this example MCL was used. The boxplot on the right hand side refers to the direct use of MCL. A few things are relevant in this example. First, and perhaps most importantly, PL is the method with the smallest variation. In particular, notice that both PL and MCL agree on average but MCL is significantly more volatile. Second, we note that even though the data was generated using $m = 4$, it looks hard to reject that m could be 3. Both sequential methods point, on average, to a parsimonious choice by saying the 3 or 4 are essentially indistinguishable. In this example, the MCMC approach seems too confident about $m = 4$ which raises, in our view, concerns regarding potential biases from using correlated samples in the approximation of marginal likelihoods. Sequential schemes offer a direct Monte Carlo approximation and PL provides improvements over MCL in delivering estimates with smaller variance.

4.2 The DP Mixture of Multivariate Normals

We now focus on the DP mixture of multivariate Normals from Example 2 in Section 1. Four datasets were simulated with dimension (d) of 2, 5, 10, and 25, and of size d times 500. In each case, for $t = 1, \ldots, T = 500d$, the d-dimensional y_t was generated from a $\mathrm{N}(\mu_t, \mathrm{AR}(0.9))$ density, where $\mu_t \overset{ind}{\sim} G_\mu$ and AR(0.9) denotes the correlation matrix implied by an autoregressive process of lag one and correlation 0.9. The mean distribution, G_μ, is the realization of a $\mathrm{DP}(4, N(0, 4\mathrm{I}))$ process. Thus the simulated data is clustered around a set of distinct means, and highly correlated within each cluster. Note that this data is similar to that used in the simulation study of Blei and Jordan (2006), but that we have the size of the

dataset change with dimension so the posterior does not become unreasonably diffuse. The DP-MVN model in (7), with fixed parametrization $\alpha = 2$, $\lambda = 0$, $\kappa = 0.25$, $\nu = d + 2$, and $\Omega = (\nu - 0.5(d + 1))\mathrm{I}$, was fit to this data. As an illustration, Figure 2 shows the data and bivariate density estimate for $d = 2$ and PL fit with $N = 1000$ particles. Here, and in the simulation study below, density estimates are the mean Rao-Blackwellized posterior predictive $\mathrm{p}(y|\mathbf{s}_T, \mathbf{n}_T, m_T)$; hence, the posterior expectation for $f(y; G)$. Marginal estimates are just the appropriate marginal density derived from the mixture of Student's t distributions in (8).

For the full simulation study, the PL algorithm with $N = 500$ particles was fit ten times to random re-orderings of the different datasets. For comparison, we also fit the same DP-MVN model with 1000 iterations (including 500 burn-in) of a Rao-Blackwellized collapsed Gibbs sampler MCMC cycling through draws of $\mathrm{p}(k_t|\mathbf{n}_T^{(-t)}, \mathbf{s}_T^{(-t)}, m_T^{(-t)}, y_t)$. The numbers of particles and iterations were chosen to lead to similar computation times, thus providing a baseline for inference without having to assess convergence of the Markov chain. PL and Gibbs studies were coded in `C++` and run simultaneously in the background on a Mac Pro with 2×3.2 GHz Quad-Core Intel Xeon processors and 4GB memory.

Noting that Quintana and Newton (2000) regard the Gibbs sampler as a *gold standard* for these types of conditionally conjugate mixture models, we further optimized the Gibbs sampler by having it based on conditional sufficient statistics and building sequentially the following initial state: each observation y_t is added to the model sequentially, allocated such that k_t corresponds to the maximal probability from (9) above. This ensures that the Gibbs chain is starting from a location with high posterior probability (as for PL, the data is randomly reordered for each of the ten runs). Since the MCMC repeatedly performs the propagation step of PL, conditional on all but one observation, and it is initialized in an optimal filtered state, we expect that it will outperform any sequential algorithm. Indeed, the connection between Gibbs sampling and PL propagation helps to explain why our method works: for each particle, the propagation of uncertainty *is* the sequential version of a Gibbs step. Thus, while it is unreasonable to expect PL to do better than Gibbs, we want to be able to obtain sequential inference that does not degenerate in comparison.

PL fit results for $d = 2$ are shown in Figure 2 and results for $d = 25$ are presented in Figure 4. In each case, while there is clear evidence of Monte Carlo error, this is to be expected

with only 500 particles and the posterior predictive density estimates are consistent with the data. The results in 25 dimensions are very encouraging: even after 12,500 observations of high-dimensional data, the marginal density estimates are very accurate. Marginal density plots for the other datasets, and for the Gibbs sampler, look very similar. In order to formally compare PL to the Gibbs sampler, we recorded the mean log predictive density score on 100 left-out observations for each dataset. That is, for each posterior sample of $\boldsymbol{s}_T$, $\boldsymbol{n}_T$, and m_T, we record $\sum_{j=1}^{100} \log \left[\frac{1}{N} \sum_{i=1}^{N} \mathrm{p}\left(\tilde{y}_j | (\mathbf{s}_t, \mathbf{n}_t, m_t)^{(i)} \right) \right]$, where $\tilde{y}_j$ is a member of the validation set and $N = 500$ for both PL and Gibbs.

The results of this comparison are shown in Figure 3. Although Gibbs is consistently better, the difference between the two methods does not appear to be widening significantly with increased dimension and time. This behavior is mirrored for the average number of allocated mixture components, an indicator of efficiency of the mixture fit. Such performance does not rely on any of the common add-on particle rejuvenation schemes. Due to the accumulation of error that will occur whenever there are not enough particles to capture fat-tail activity, we can only expect performance to be improved if the algorithm is populated by conditional sufficient statistics sampled in a previous analysis (as opposed to being populated with a single sufficient statistic). And even in 25 dimensions, the uncertainty update for a new observation requires an average of only 10 seconds (including prediction time, while sharing processors with the Gibbs sampler). As such, PL will be able to provide on-line inference for high-frequency high-dimensional data, possibly after deriving initial particles from a posterior fit provided by MCMC or (for very large training sets) variational methods. Furthermore, although the Gibbs sampler is able to outperform in terms of predictive ability, the 500 Gibbs observations are correlated while the 500 PL observations are (up to the particle approximation) independent. Finally, we note that the program used in this study executes in a standard sequential manner, even though major computational gains could be made by taking advantage of the inherently parallel nature of PL.

Learning Hyper-Parameters. Note that we can also assign hyperpriors to the parameters of G_0. In this case, a parameter learning step for each particle is added to the algorithm. This entails an expansion of the essential state vector to now include particles of λ and Ω. Assuming

a $\mathrm{W}(\gamma_\Omega, \Psi_\Omega^{-1})$ prior for Ω and a $\mathrm{N}(\gamma_\lambda, \Psi_\lambda)$ prior for λ, the sample at time t is augmented with draws for the auxiliary variables $\{\mu_j^\star, \Sigma_j^\star\}$, for $j = 1, \ldots, m_t$, from their posterior full conditionals, $\mathrm{p}(\mu_j^\star, \Sigma_j^\star | \mathbf{s}_t, \mathbf{n}_t) = \mathrm{N}(\mu_j^\star;\ a_{t,j}, \frac{1}{\kappa+n_{t,j}}\Sigma_j^\star)\mathrm{W}(\Sigma_j^{\star-1};\ \nu + n_{t,j}, \Omega + D_{t,j})$. The parameter updates are then

$$\lambda \sim \mathrm{N}\left(R(\gamma_\lambda \Psi_\lambda^{-1} + \kappa \sum_{j=1}^{m_t} \Sigma_j^{\star-1}\mu_j^\star),\ R \right) \quad \text{and} \quad \Omega \sim \mathrm{W}\left(\gamma_\Omega + m_t\nu, R^{-1}\right), \tag{30}$$

where $R^{-1} = \sum_{j=1}^{m_t} \Sigma_j^{\star-1} + \Psi_\Omega^{-1}$.

Similarly, if we wish to learn about the concentration parameter α from an usual gamma hyperprior, the essential state vector is further augmented to include particles for α and the auxiliary variable method from Escobar and West (1995) can be used in the learning step. Figure 5 shows prior to posterior learning for α in a $d = 1$ simulation study.

The ability to augment the essential state vector allow the user of PL to deal with a variety of non conjugate model specifications as well as deal with the problem of learning hyperparameters of complex mixture models.

4.3 Inference about the Random Mixing Distribution

All of the inference in Section 4.2 is based on the marginal posterior predictive, thus avoiding direct simulation of the infinite dimensional random mixing distribution. In some situations, however, it is necessary to obtain inference about the actual posterior for the random density $f(y; G)$, and hence about G itself, rather than about $\mathbb{E}[f(y; G)]$. For example, functionals of the conditional density $f(x, y; G)/f(x; G)$ are the objects of inference in implied conditional regression (e.g., Taddy and Kottas, 2009), and Kottas (2006) describes inference for the hazard function derived from $f(y; G)$. The standard approach to sampling G is to apply a truncated version of the constructive definition in (21) to draw from $\mathrm{DP}(\alpha + t, G_0^t(\theta; \mathbf{n}_t, \boldsymbol{\theta}_t^\star))$, the conjugate posterior for G given $\boldsymbol{\theta}_t^\star$ and $\mathbf{n}_t$ (refer to Gelfand and Kottas (2002) for truncation guidelines and posterior consistency results). The approximate posterior draw $G_L \equiv \{p_l, \vartheta_l\}_{l=1}^L$ is built from i.i.d. point mass locations $\vartheta_l \sim G_0^t(\vartheta_l; \mathbf{n}_t, \boldsymbol{\theta}_t^\star)$, defined as in (22), and the probability vector $\boldsymbol{p} = (p_1, \ldots, p_L)$ from the finite stick- breaking process $p_l = v_l(1 - \sum_{j=1}^{l-1})$ for $l = 1, \ldots, L$, with $v_l \sim \mathrm{beta}(1, \alpha + t)$ and $v_L = 1$.

As in the finite mixtures case, we are able to Rao-Blackwellized inference and sample the parameters of interest (in this case, G_L), conditional on each sampled set of sufficient statistics. In particular,

$$\begin{aligned} \mathrm{p}(G_L|\mathbf{s}_t, \mathbf{n}_t, m_t) &= \mathrm{p}(\boldsymbol{p}, \boldsymbol{\vartheta}|\mathbf{s}_t, \mathbf{n}_t, m_t) = \int \mathrm{p}(\boldsymbol{p}, \boldsymbol{\vartheta}|\boldsymbol{\theta}_t^\star, \mathbf{n}_t) d\mathrm{P}(\boldsymbol{\theta}_t^\star|\mathbf{n}_t, \mathbf{s}_t, m_t) \\ &= \mathrm{sb}_L(\boldsymbol{p}; \mathrm{beta}(1, \alpha+t)) \int \prod_{l=1}^{L} dG_0^t(\vartheta_l; \boldsymbol{\theta}_t^\star, \mathbf{n}_t) d\mathrm{P}(\boldsymbol{\theta}^\star|\mathbf{n}_t, \mathbf{s}_t, m_t) \\ &= \mathrm{sb}_L(\boldsymbol{p}; \mathrm{beta}(1, \alpha+t)) \prod_{l=1}^{L} \left[\frac{\alpha}{\alpha+t} dG_0(\vartheta_l) + \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha+t} \mathrm{p}_{\theta_j^\star}(\vartheta_l|n_{t,j}, s_{t,j}) \right], \end{aligned} \tag{31}$$

where $\mathrm{p}_{\theta_j^\star}(\cdot|n_{t,j}, s_{t,j})$ is the posterior full conditional for $\theta_j^\star$ given $(n_{t,j}, s_{t,j})$. For example, in a draw from the posterior for G_L in the DP-MVN model of Section 4.2, the component locations $\vartheta_l = [\mu_l, \Sigma_l]$ are sampled i.i.d. from $\mathrm{p}(\vartheta_l|k_l = j, n_{t,j}, s_{t,j}) = \mathrm{N}(\mu_l;\ a_{t,j}, \frac{1}{\kappa+n_{t,j}}\Sigma_l)\mathrm{W}(\Sigma_l^{-1}; \nu + n_{t,j}, \Omega + D_{t,j})$ with probability $n_{t,j}/(\alpha+t)$ for $j = 1, \ldots, m_t$, and from $G_0(\mu_l, \Sigma_l) = \mathrm{N}(\mu_l; \lambda, \Sigma_l/\kappa)$ $\mathrm{W}(\Sigma_l^{-1}; \nu, \Omega)$ with probability $\alpha/(\alpha+t)$, and component weights are just drawn according to the appropriate stick-breaking construction.

We illustrate this approach with the conditional inference that results from an application to data generated such that $x_t \sim \mathrm{N}(0, 1)$, $y_t = 0.3 + 0.4x_t + 0.5\sin(2.7x_t) + 1.1(1+x_t^2)^{-1} + \varepsilon_t$, where $\varepsilon_t \sim N(0, \sigma_t^2)$ such that $\sigma_t = 0.5$ with probability $\Phi(x_t)$ and $\sigma_t = 0.25$ with probability $1 - \Phi(x_t)$. This data corresponds to a nonlinear mean plus heteroskedastic additive error, and was previously used as test function in Taddy and Kottas (2009). The joint distribution of x and y is modeled as arising from DP-MVN of Section 4.2, with the added parameter learning of equation (30), parametrized by $\alpha = 2$, $\nu = 3$, $\kappa = 0.1$, $\gamma_\lambda = 0$, $\Psi_\lambda = 1.5\mathrm{I}$, $\gamma_\Omega = 3$, and $\Psi_\Omega = 0.1\mathrm{I}$. After applying PL with $N = 1000$ particles to filter the posterior, truncated approximations G_L with $L = 300$ were drawn as in (31), and given these draws conditional inference follows the techniques detailed in Taddy and Kottas (2009). In particular, the conditional density is available at any location (x, y) as $\sum_{l=1}^{L} p_l \mathrm{N}(x, y; \mu_l, \Sigma_l) / f(x; G_L)$, and the conditional mean at x is $\mathbb{E}[Y|x; G_L] = \sum_{l=1}^{L} p_l \mathrm{N}(x; \mu_l^x, \sigma_l^x) [\mu_l^y + \rho_l^{xy}(\sigma_l^x)^{-1}(x - \mu_l^x)] / f(x; G_L)$, where $f(x; G_L) = \sum_{l=1}^{L} p_l \mathrm{N}(x; \mu_l^x, \sigma_l^x)$, $\mu = (\mu^x, \mu^y)$, and Σ is partitioned with diagonal (σ^x, σ^y) and off-diagonal ρ^{xy}. Figure 6 shows the results of our analysis, and it appears that the Rao-Blackwellized G_L sample is able to capture the conditional relationship (even at the bound-

aries). As a final remark, it is appealing that, as opposed to blocked-Gibbs schemes, we only need to draw G_L after obtaining a filtered posterior for conditional sufficient statistics, thus allowing us to choose the truncation L based on properties of the posterior (and providing a Rao-Blackwellized version of the approach in Gelfand and Kottas (2002)).

5 Conclusion

We have proposed a new estimation method for general mixture models. A vast body of empirical and theoretical evidence of the robust behavior of the resample/propagate PL procedure in states space models appear in Carvalho et al. (2010) and in more general contexts in Lopes et al. (2010) and following discussion. Additionally, conditioning on sufficient statistics for states and parameters whenever possible creates a Rao-Blackwellized filter with more uniformly distributed resampling weights. Finally, PL does not attempt to approximate the ever increasing joint posterior distribution for k^t. It is self evident that any importance sampling approximation to the entire vector of allocations will eventually fail, due to the curse of dimensionality, as t grows. But we show that this is an irrelevant target, since the allocation problem can be effectively solved *after* filtering relevant sufficient information. Finally, we include an efficient framework for marginal likelihood estimation, providing a valuable tool for real-time sequential model selection.

The approach is easy to understand, simple to implement, and computationally fast (standard implementation of PL for mixtures is available in the `R` package `Bmix`). The framework is especially appealing in the large class of nonparametric mixture priors where the predictive probability function is either available analytically or possible to approximate. To enable understanding, we have focused on a limited set of concrete models, while pointing to a more general applicability available with little change to the algorithm. It is thus hoped that this article will facilitate a wider adoption of sequential particle methods in nonparametric mixture model applications.

References

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics 2*, 1152 – 1174.

Basu, S. and S. Chib (2003). Marginal Likelihood and Bayes Factors for Dirichlet Process Mixture Models. *Journal of the American Statistical Association 98*, 224–235.

Blackwell, D. and J. MacQueen (1973). Ferguson distributions via Pólya urn schemes. *Annals of Statistics 1*, 353–355.

Blei, D. M. and M. I. Jordan (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis 1*(1), 121–144.

Bush, C. and S. MacEachern (1996). A semiparametric Bayesian model for randomized block designs. *Biometrika 83*, 275–285.

Carvalho, C. M., M. Johannes, H. F. Lopes, and N. Polson (2010). Particle learning and smoothing. *Statistical Science 25*, 88-106.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association 90*, 1313–1321.

Chib, S. and I. Jeliazkov (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association 96*, 270–281.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika 89*, 539–551.

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, B 68*, 411–436.

Dunson, D. B. and J.-H. Park (2008). Kernel stick-breaking processes. *Biometrika 95*, 307–323.

Escobar, M. and M. West (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association 90*, 577–588.

Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing 14*, 11–21.

Fearnhead, P. and Meligkotsidou, L. (2007). Filtering methods for mixture models *Journal of Computational and Graphical Statistics 6*, 586–607.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics 1*, 209–230.

Ferguson, T. S. (1974). Prior distributions on spaces of probability measures. *Annals of Statistics 2*, 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normals distributions. In M. RIzvi, J. Rustagi, D. Siegmund (Eds.), *Recent Advances in Statistics*, New York Academic Press.

Gelfand, A. E. and A. Kottas (2002). A computational approach for full nonparametric Bayesian inference under Dirichlet process mixture models. *Journal of Computational and Graphical Statistics 11*, 289–305.

Gelfand, A. E., A. Kottas, and S. N. MacEachern (2005). Bayesian nonparametric spatial modeling with Dirichlet process mixing. *Journal of the American Statistical Association 100*, 1021–1035.

Godsill, S. J., A. Doucet, and M. West (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association 99*, 156–168.

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

Griffiths, T. L. and Z. Ghahramani (2006). Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems*, Volume 18.

Han, C. and B. Carlin (2001). Markov chain Monte Carlo methods for computing Bayes factors: A comparative review. *Journal of the American Statistical Association 96*, 161–173.

Ishwaran, H. and L. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96*, 161–173.

Ishwaran, H. and M. Zarepour (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika 87*, 371–390.

Jasra, A., Stephens, D. and Holmes, C. (2007). On Population-based simulation for static inference. *Statistics and Computing 17*, 263–279.

Kong, A., J. S. Liu, and W. H. Wong (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association 89*, 278–288.

Kottas, A. (2006). Nonparametric Bayesian survival analysis using mixtures of Weibull distributions. *Journal of Statistical Planning and Inference 136*, 578–596.

Lee, J., F. A. Quintana, P. Müller, and L. Trippa (2008). Defining predictive probability functions for species sampling models. Working Paper.

Liu, J. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association 93*, 1032–1044.

Lo, A. Y. (1984). On a class of Bayesian nonparametric estimates. I: Density Estimates. *Annals of Statistics 12*, 351–357.

Lopes, H., C. M. Carvalho, M. Johannes and N. Polson. Particle Learning for Sequential Bayesian Computation (with discussion). In J. Bernardo, M. J. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. F. M. Smith and M. West (Eds.), *Bayesian Statistics*, Volume 9. Oxford. *In Press.*

MacEachern, S. N., M. Clyde, and J. S. Liu (1999). Sequential importance sampling for nonparametric Bayes models The next generation. *The Canadian Journal of Statistics 27*, 251–267.

MacEachern, S. N. and P. Müller (1998). Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics 7*, 223–238.

McKenzie, E. (1985). An autoregressive process for beta random variables. *Management Science 31*, 988–997.

Muliere, P. and S. Walker (1997). A Bayesian non-parametric approach to survival analysis using Pólya trees. *Scandinavian Journal of Statistics 24*, 331–340.

Müller, P. and F. A. Quintana (2004). Nonparametric Bayesian data analysis. *Statistical Science 19*, 95–110.

Neal, R. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics 9*, 249–265.

Paddock, S. M., F. Ruggeri, M. Lavine, and M. West (2003). Randomized Pólya tree models for nonparametric Bayesian inference. *Statistica Sinica 13*, 443–460.

Perman, M., J. Pitman, and M. Yor (1992). Size-biased sampling of Poisson point processes and excursions. *Probability Theory and Related Fields 92*, 21–39.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields 102*, 145–158.

Pitt, M. and N. Shephard (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association 94*, 590–599.

Quintana, F. and M. A. Newton (2000). Computational aspects of nonparametric bayesian analysis with applications to the modelling of multiple binary sequences. *Journal of Computational and Graphical Statistics 9*, 711–737.

Regazzini, E. (1998). Old and recent results on the relationship between predictive inference and statistical modelling either in nonparametric or parametric form. In J. Bernardo, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics*, Volume 6. Oxford.

Richardson, S. and P. J. Green (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society - Series B 59*, 731–792.

Rodriguez, A. and D. B. Dunson (2009). Nonparametric Bayesian models through probit stick-breaking processes. Technical Report UCSC-SOE-09-12, University of California Santa Cruz.

Rodriguez, A., D. B. Dunson, and A. E. Gelfand (2008). The nested Dirichlet process. *Journal of the American Statistical Association 103*, 1131–1154.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B, Methodological 62*(4), 795–809.

Taddy, M. A. (2010). Autoregressive mixture models for dynamic spatial Poisson processes: Application to tracking intensity of violent crime. *Journal of the American Statistical Association to appear.*

Taddy, M. and A. Kottas (2009). Bayesian nonparametric approach to inference for quantile regression. *Journal of Business and Economic Statistics*. To Appear.

Walker, S. G., P. Damien, P. W. Laud, and A. F. M. Smith (1999). Bayesian nonparametric inference for random distributions and related functions. *Journal of the Royal Statistical Society, Series B 61*, 485–527.

Walker, S. G. and P. Muliere (1997). Beta-Stacy processes and a generalization of the Pólya-urn scheme. *The Annals of Statistics 25*, 1762–1780.

West, M. and J. Harrison (1997). *Bayesian Forecasting and Dynamic Models* (2nd ed.). Springer Series in Statistics. Springer.

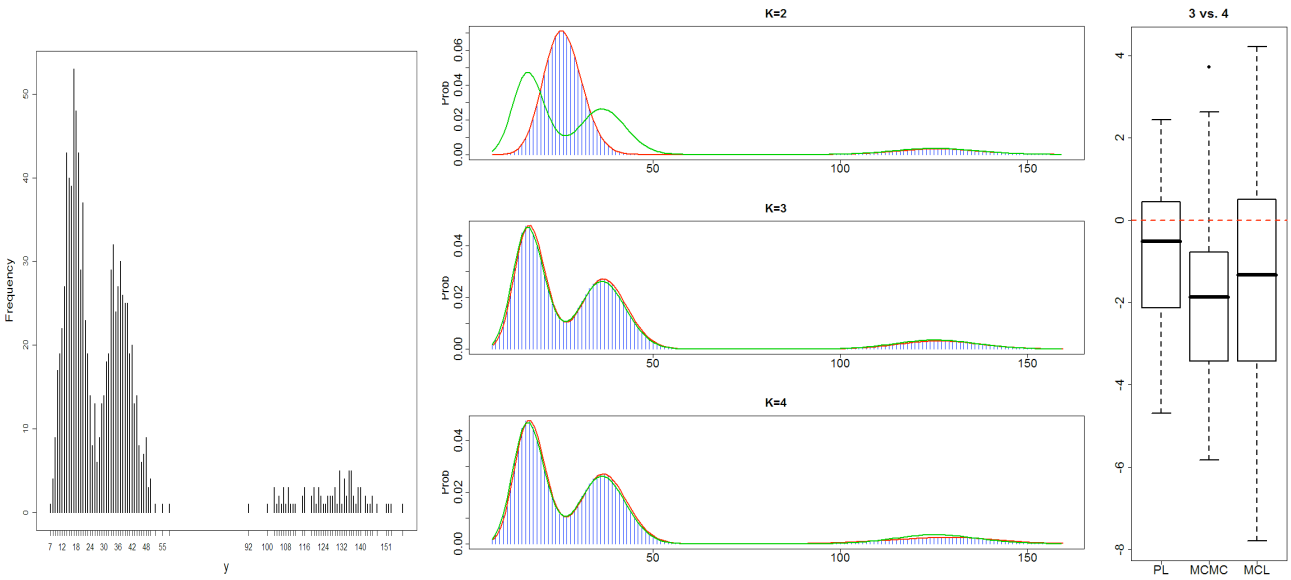

Figure 1: **Poisson Mixture Example.** Left Panel: data from a $m = 4$ mixture of Poisson. Central Panel: density estimates from PL (red), MCMC (Blue) and the true density (green). Right Panel: MC study of Bayes factors for comparing $m = 3$ vs. $m = 4$.

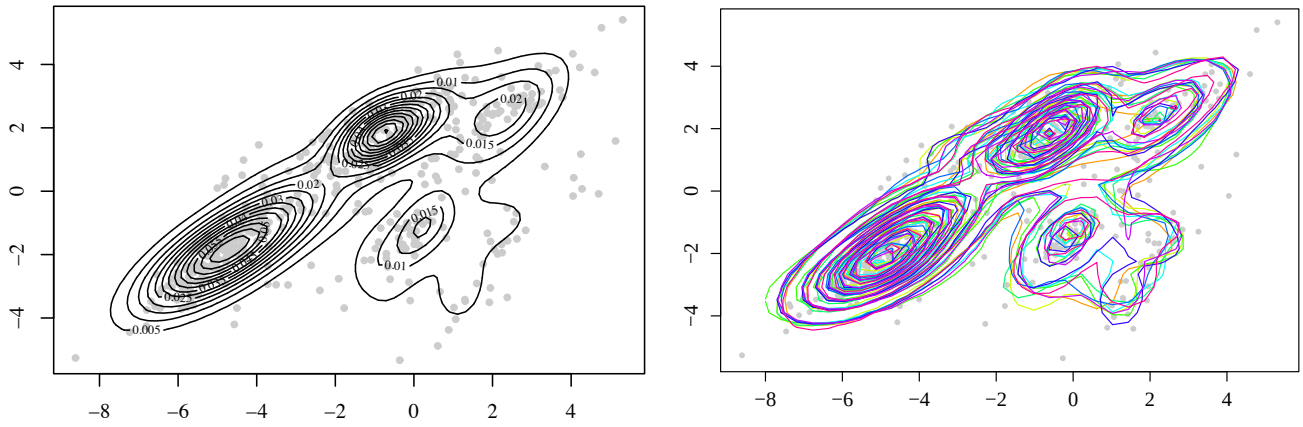

Figure 2: **DP-MVN Example.** Data and density estimates for PL fit with 1000 particles (left) and each of ten PL fits with 500 particles (right), to a random ordering of the 1000 observations of 2 dimensional data generated as in Section 4.2.

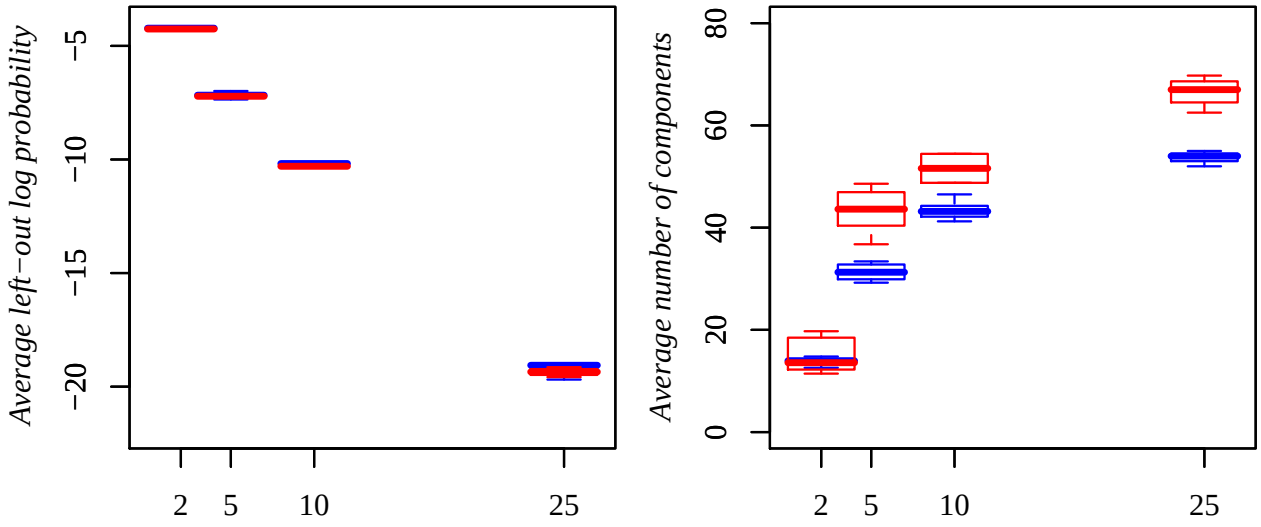

Figure 3: **DP-MVN Example.** Results from the simulation study in Section 4.2. The left plot shows average log posterior predictive score for validation sets of 100 observations, and the right plot shows the posterior averages for m_T, the total number of allocated mixture components. In each case, boxplots illustrate the distribution over ten repetitions of the algorithm. Red boxplots correspond to PL, and the blue correspond to Gibbs.

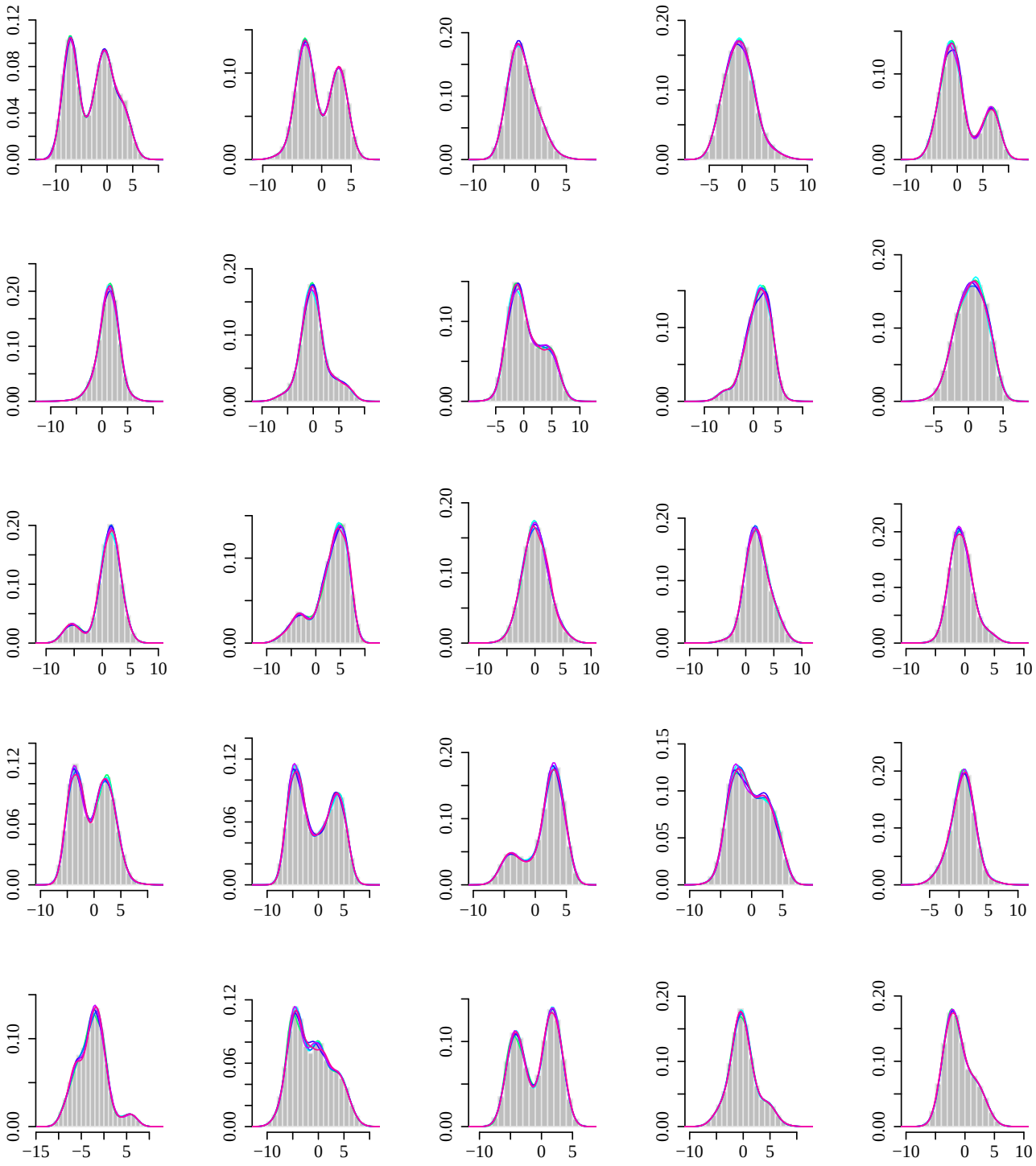

Figure 4: **DP-MVN Example.** Data and marginal density estimates for the DP-MVN model fit to 12,500 observations of 25 dimensional data. The colors represent mean posterior estimates for each of ten PL fits, with 500 particles, to a random ordering of the data.

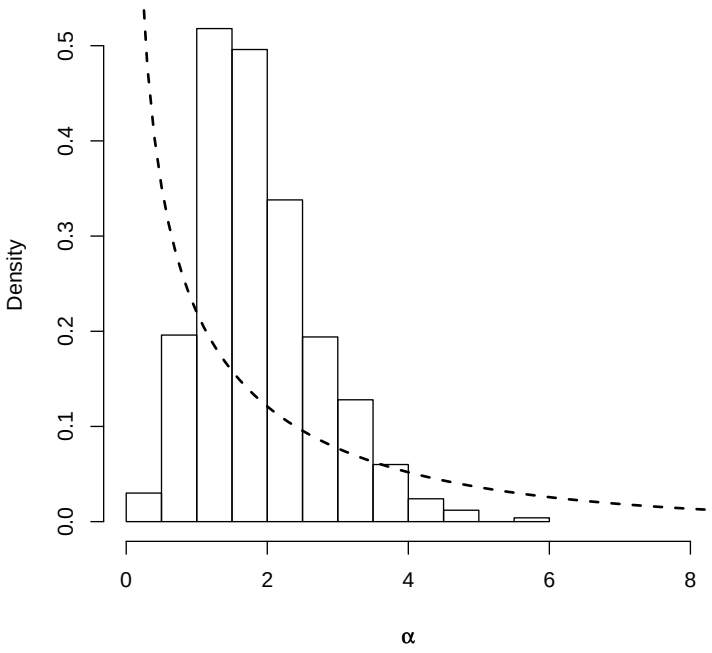

Figure 5: Histogram approximation to the posterior for the concentration parameter α in a DP mixture of univariate normals (applied to to the well studied Galaxy Data). The dotted line refers to the prior $\alpha \sim Ga(0.5, 0.25)$.

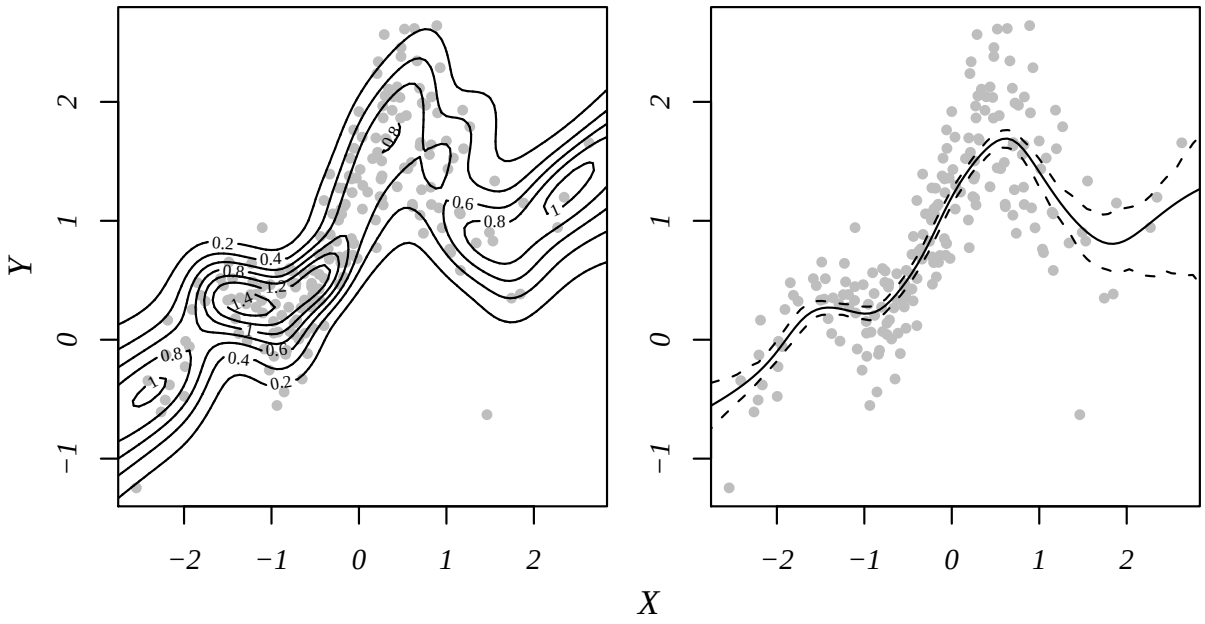

Figure 6: **Regression Example.** Data and conditional inference for the example of Section 4.3. The left panel shows the filtered posterior mean estimate for the conditional density $f(x, y; G_L)/f(x; G_L)$, and the right panel shows the posterior mean and 90% interval for the mean $\mathbb{E}[y|x; G_L]$.