Sequential Monte Carlo Samplers for Dirichlet Process Mixtures

Yener Ulker and Bilge Gunsel
Dept. of Electronics and Communications Eng.
Istanbul Technical University
34469 Maslak Istanbul, Turkey
yenerulker@itu.edu.tr, gunselb@itu.edu.tr

A. Taylan Cemgil
Dept. of Computer Engineering,
Bogazici University
34342 Bebek Istanbul, Turkey
taylan.cemgil@boun.edu.tr

Abstract

In this paper, we develop a novel online algorithm based on the Sequential Monte Carlo (SMC) samplers framework for posterior inference in Dirichlet Process Mixtures (DPM) (DelMoral et al., 2006). Our method generalizes many sequential importance sampling approaches. It provides a computationally efficient improvement to particle filtering that is less prone to getting stuck in isolated modes. The proposed method is a particular SMC sampler that enables us to design sophisticated clustering update schemes, such as updating past trajectories of the particles in light of recent observations, and still ensures convergence to the true DPM target distribution asymptotically. Performance has been evaluated in a Bayesian Infinite Gaussian mixture density estimation problem and it is shown that the proposed algorithm outperforms conventional Monte Carlo approaches in terms of estimation variance and average log-marginal likelihood.

1 Introduction

In recent years, Dirichlet Process Mixtures (DPM) model (Antoniak, 1974) has been one of the most widely used and popular approach to nonparametrical probabilistic models. Originally, DPM have been widely used as a building block in hierarchical models for solving density estimation and clustering problems where the actual form of the data generation process is not constrained to a particular parametric family.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010. A. Taylan Cemgil is supported by Bogazici University research fund BAP 09A105P

Provided that inference can be carried out effectively for the DPM, at least in principle, any density can be approximated with arbitrary precision. However, exact inference is unfortunately intractable. Yet due to the mentioned potential advantages of nonparametric approaches, there has been a surge of interest to the DPM model and efficient inference strategies based on variational techniques (Blei and Jordan, 2004) and Monte Carlo Markov Chain (MCMC) (Escobar and West, 1992; Jain and Neal, 2000).

By construction, the DPM model is exchangable and the ordering of data does not matter. However, for inference it is nevertheless beneficial to process data sequentially in some natural order. Such an approach gives computational advantages especially for large datasets. In the literature a number of online inference techniques have been proposed to estimate an artificially time evolving DPM posterior (MacEachern et al., 1999; Quintana, 1996; Fearnhead, 2004).

However, it is argued that sequential importance sampling is not an appropriate method for models with static parameters and especially large datasets due to the degeneracy phenomenon and accumulated Monte Carlo error over time (Quintana and Newton, 1998). The sampler becomes 'sticky', meaning that previously assigned clusterings can never be updated according to the information provided by the latest observations. In order to overcome this drawback, we propose an efficient sequential Monte Carlo sampler that estimates the sequentially evolving DPM model posterior. Unlike the existing methods (MacEachern et al., 1999; Quintana, 1996; Fearnhead, 2004), our algorithm enables us to update past trajectories of the particles in the light of recent observations. The proposed method takes advantage of the SMC sampler framework to design such update schemes (DelMoral et al., 2006).

2 Dirichlet Process Mixtures (DPM)

In a batch Bayesian setting, the joint distribution corresponding to a finite mixture model over N observations $y = \{y_i\}$, $i = 1 \dots N$, can be defined as follows:

$$p(\theta, z, y) = p(z) \left(\prod_{i=1}^{N} g(y_i|\theta_{z_i}) \right) \prod_{j=1}^{k} p(\theta_j). \quad (1)$$

Here, for $i = 1 \dots N$, $z_i \in \{1 \dots k\}$ denotes the cluster index of the i th observation and $\theta = \{\theta_j\}, j \in \{1 \dots k\}$ denote the cluster conditional parameters where k refers to the maximum number of clusters. We will use $z = \{z_i\}, i = 1 \dots N$ to refer to clustering variables, that we also call cluster labels or simply labels.

However, the number of mixture components is often unknown in practice and the DPM model provides an elegant solution for construction of mixture models with unknown number of components. In the sequel, we will refer to the target posterior as

$$\pi(x) \equiv p(z, \theta|y) \quad (2)$$

where $x = \{z, \theta\}$. It is advantageous to construct a mixture model sequentially, where data arrives one by one. To denote the sequential construction, we extend our notation with an explicit 'time' index n.

We denote the observation sequence at time n by $y_n = \{y_{n,1} \dots y_{n,n}\}$. Each observation $y_{n,i}$, $i = 1, \dots n$, is assigned to a cluster where $z_{n,i} \in \{1, \dots k_n\}$ is the cluster label and, $k_n \in \{1 \dots n\}$ represent the number of clusters at time n. The vector of cluster variables is defined as $z_n = \{z_{n,1} \dots z_{n,n}\}$ and corresponding cluster parameters are represented with the parameter vector $\theta_n = \{\theta_{n,1} \dots \theta_{n,k_n}\}$.

The DPM model assumes that the cluster parameters are independently drawn from the prior $\pi(\theta)$ and the observations are independent of each other conditional on the assignment variable $z_{n,i}$. Hence the DPM posterior density $\pi_n(x_n)$ can be expressed as,

$$\pi_n(x_n) \propto p(z_n) \prod_{j=1}^{k_n} p(\theta_{n,j}) \prod_{i=1}^{n} g(y_{n,i}|\theta_{n,z_{n,i}}) \quad (3)$$

where $x_n = \{z_n, \theta_n\}$. The prior on clustering variable vector z_n is formulated by Eq.(4) in a recursive way,

$$p(z_{n,i+1} = j | z_{n,\{1:i\}}) = \begin{cases} \frac{l_j}{i+\kappa}, & j = 1, \dots, k_i \\ \frac{\kappa}{i+\kappa}, & j = k_i + 1 \end{cases} \quad (4)$$

where k_i is the number of clusters in the assignment $z_{n,\{1:i\}}$. l_j is the number of observations that $z_{n,\{1:i\}}$ assigns to cluster j and κ is a 'novelty' parameter.

In our work, we assume that a conjugate prior is chosen such that given z_n, the parameter θ_n can be integrated out and the DPM posterior distribution can be calculated up to a normalizing constant.

3 Sequential Monte Carlo (SMC) Samplers

Sequential inference schemes have limited success in maintaining an accurate approximation to the true target density. Particularly for large datasets, Monte Carlo error accumulates over time and the estimation variance increases (Quintana and Newton, 1998). This is due to the fact that past states of particle trajectories (i.e., past clusterings) are not updated with new observations. The problem can be alleviated by a retrospective method that is able to reassign the previous clusterings $\{z_{n,1} \dots z_{n,n-1}\}$ at time n according to latest observations received. The SMC samplers framework enables us to accomplish this in practice and still ensures convergence to the true target posterior asymptotically (DelMoral et al., 2006).

3.1 The Method

In sequential Monte Carlo algorithms such as particle filtering, we sample from a sequence of target densities evolving with a countable index n, $\pi_1(x_1) \dots \pi_n(x_n)$, each defined on a common measurable space $(E_n, \mathcal{E}_n)$ where $x_n \in E_n$. Conventionally the particle filter is defined on the sequence of target densities $\pi_1(x_1) \dots \pi_n(x_n)$ where corresponding proposal distributions are defined as $\eta_1(x_1) \dots \eta_n(x_n)$. The unnormalized importance weight w_n at time n can be defined as,

$$w_n = \frac{\gamma_n(x_n)}{\eta_n(x_n)} \quad (5)$$

where γ_n is the unnormalized target distribution according to $\pi_n = \gamma_n / Z$, and Z is the normalizing constant. In order to derive the importance weights sequentially one needs to calculate the proposal distribution $\eta_n(x_n)$ pointwise.

Computation of the importance distribution $\eta_n(x_n)$ for $n > 1$ requires an integration with respect to $x_{1:n-1} = \{x_1 \dots x_{n-1}\}$ thus a closed form solution to $\eta_n(x_n)$ is not available except for specifically designed kernels such as $K(x_{n-1}, x_n) = K(x_n)$. SMC samplers (DelMoral et al., 2006) circumvent this limitation using an auxiliary variable technique which solves the sequential importance sampling problem in an extended space $E^n = \{E_1 \times \dots \times E_n\}$. One performs importance sampling between the joint importance distribution $\eta_n(x_{1:n})$ and the artificial joint target distribution

defined by $\widetilde{\pi}_n(x_{1:n}) = \widetilde{\gamma}_n(x_{1:n})/Z_n$ where Z_n denotes the normalizing constant. The algorithm enables us to calculate efficient weight update equations for a huge class of proposal kernels $K_n(x_{n-1}, x_n)$, such as MCMC transition kernels.

The proposal distribution $\eta_n(x_{1:n})$ of SMC is defined on the extended space E^n as follows,

$$\eta_n(x_{1:n}) = \eta_1(x_1) \prod_{k=2}^{n} K(x_{k-1}, x_k). \tag{6}$$

Note that here an integration is no longer required. However, this comes with the expense of an artificial target density that is also defined on a larger space:

$$\widetilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n) \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k). \tag{7}$$

Here, a sequence of backward kernels $L_k(x_{k+1}, x_k)$, $k = \{1 \dots n-1\}$ is introduced to define the artificial target distribution shown in Eq.(7). By construction, the joint posterior $\widetilde{\pi}_n(x_{1:n})$ defined on the extended space E^n admits $\pi_n(x_n)$ as a marginal. Therefore the resultant weight function ensures convergence to the original target density. The generic SMC algorithm which is used to sample from a sequentially evolving target posterior $\widetilde{\pi}_n$ is presented as follows:

Assume that a set of weighted particles $\left\{W^i_{n-1}, X^i_{1:n-1}\right\}_{i=1}^{N_p}$ approximate $\widetilde{\pi}_{n-1}$ at time $n-1$. At time n the path of each particle can be extended using a Markov kernel, $K_n(x_{n-1}, x_n)$. The unnormalized importance weights associated with the extended particles are calculated according to Eq.(8),

$$\begin{aligned} w_n(x_{1:n}) &= w_{n-1}(x_{1:n-1}) v_n(x_{n-1}, x_n) \\ &= \frac{\widetilde{\gamma}_n(x_{1:n})}{\eta_n(x_{1:n})} \end{aligned} \tag{8}$$

where the incremental term of weight equation, $v_n(x_{n-1}, x_n)$, is equal to

$$v_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n) L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}. \tag{9}$$

The discrepancy between η_n and $\widetilde{\gamma}_n$ tends to grow with n, consequently the variance of the unnormalized importance weights increases. A resampling step is used if the variance is above a certain level as measured by, e.g, effective sample size (ESS).

3.2 Backward kernels

Design of efficient sampling schemata hinges on properly choosing L_n with respect to K_n. The introduction of the L_n extends the integration domain from E to E^n and eliminates the necessity of calculating $\eta_n(x_n)$. However increasing the integration domain also increases the variance of the importance weights. In (DelMoral et al., 2006) it is shown that the optimal backward Markov kernel L^{opt}_{k-1} $(k = 2, \dots, n)$ minimizing the variance of the unnormalized importance weight $w_n(x_{1:n})$ is given for any k by,

$$L^{opt}_{k-1}(x_k, x_{k-1}) = \frac{\eta_{k-1}(x_{k-1}) K_k(x_{k-1}, x_k)}{\eta_k(x_k)} \tag{10}$$

However, the kernel given by Eq.(10) usually does not admit a closed form solution. Therefore common strategy is to approximate the optimal kernel as close as possible to provide asymptotically consistent estimates (DelMoral et al., 2006) . A sensible approximation at a given time step n can be obtained by substituting π_{n-1} for η_{n-1}, where the approximate kernel L_{n-1} can be expressed as in Eq.(11),

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}{\int \pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n) dx_{n-1}} \tag{11}$$

that can yield a closed form solution to the weight update equation if it is possible to calculate the integration. An alternative to approximate backward kernel can be obtained as in Eq.(12) by replacing $\pi_{n-1}(x_{n-1})$ by $\pi_n(x_{n-1})$ and selecting K_n as a MCMC kernel targeting π_n in Eq.(11).

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1}) K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \tag{12}$$

Note that, although Eq.(11) is a closer approximation to the optimal bakward kernel, Eq.(12) can lead to simpler weight update equations.

4 A SMC sampler for the Dirichlet Process Mixtures

In this section we will explain the proposed SMC based algorithm that generates weighted samples from the DPM model posterior described in Section 2.

Now, assuming conjugacy, we devise an algorithm that approximates the posterior distribution,

$$P(z_n | y_n) \approx \sum_{i=1}^{N_p} W^i_n \delta_{Z^i_n}(z_n) \tag{13}$$

with a set of weighted samples $\left\{W^i_n, Z^i_n\right\}_{i=1}^{N_p}$ where each particle Z^i_n encodes an assignment vector of all datapoints upto time n, formally represented with a Dirac delta function $\delta_{Z^i_n}(z_n)$.

Let us define a forward kernel, K_n, generating samples from the sequence of distributions built according to

Eq.(3). We first partition an assignment vector $z_n = \{z_{n,r}, z_{n,d}, z_{n,n}\}$ where r is a subset of $\{1,\ldots,n-1\}$, a set of not necessarily consecutive indicies, and $d = \{1,\ldots,n-1\} - r$. Note that throughout the text we will call the set $z_{n,r}$ as the active block. We define $u = r \cup \{n\}$, and denote $-u \equiv d$. Exploiting the conjugacy property, we propose using the following conditional distribution for K_n as given in Eq.(14).

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1,-u}}(z_{n,-u})\, \pi_n(z_{n,u}|z_{n,-u}) \quad (14)$$

This kernel allows us updating $z_{n,u}$ which includes the current and some past assignments without changing the rest $z_{n,-u}$.

By replacing K_n in Eq.(11) we obtain the straight derivation to the approximate kernel,

$$L_{n-1}(z_n, z_{n-1}) = \delta_{z_{n,-u}}(z_{n-1,-u}) \times \pi_{n-1}(z_{n-1,r}|z_{n-1,-r}). \quad (15)$$

Given our choices of the forward and backward kernels, now we are able to write the expression for the incremental weight function given in Eq.(9) as follows,

$$v_n(z_{n-1}, z_n) = \frac{\gamma_n(z_{n,-u})}{\gamma_{n-1}(z_{n-1,-r})}. \quad (16)$$

The proposed scheme can also be seen as a generalization of a conventional particle filtering weight update scheme. The particle filter simply uses the forward kernel $K_n(z_{n-1}, z_n) = \delta_{z_{n-1}}(z_{n,-n})\pi_n(z_{n,n}|z_{n,-n})$. In this case only the clustering variable $z_{n,n}$ is updated upon arrival of the new observation that yields the weighting function given in Eq.(17).

$$v_n^{\mathrm{pf}}(z_{n-1}, z_n) = \frac{\gamma_n(z_{n,-n})}{\gamma_{n-1}(z_{n-1})} \quad (17)$$

The sequential imputation scheme (Liu, 1996) and many particle filtering based methods (Quintana and Newton, 1998; Chen and Liu, 2000) use the simplified incremental weight update function given by Eq.(17). Note that such a kernel selection strategy is not capable of updating the active set $z_{n,r}$ according to the new observations, therefore can yield to poor estimation performance.

In order to render our sampling approach more efficient by making more global moves we wish to change a block of variables, i.e., choose the cardinality of the index set r as large as possible. However, when the cardinality of r increases, the time required for the exact computation of the incremental weight in Eq.(16) grows exponentially. In the sequel, we will define MCMC and approximate Gibbs type moves where the associated weight update equations can be computed efficiently. This leads to low complexity algorithms for sampling from the time evolving DPM posterior distribution.

4.1 MCMC kernels

We first define the forward kernel as

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1,-u}}(z_{n,-u}) \times K_n(z_{n,n}, z_{n,r}|z_{n-1}) \quad (18)$$

where $K_n(z_{n,n}, z_{n,r}|z_{n-1})$ is a valid MCMC kernel applying a single Gibbs step targeting the full conditional distribution $\pi_n(z_{n,n}, z_{n,r}|z_{n,-u})$. Intuitively, this kernel updates the active block using a Gibbs sampler and constructs the proposal distribution using the sequence of full conditional distributions.

A corresponding backward kernel can be obtained by substituting $K_n(z_{n-1}, z_n)$ into the Eq.(12). This yields in the following incremental weight update equation,

$$v_n^{gb}(z_{n-1}, z_n) = \frac{\gamma_n(z_{n-1,r}, z_{n,-u})}{\gamma_{n-1}(z_{n-1})}. \quad (19)$$

Note that as a consequence of the chosen backward kernel, Eq.(19) is independent from the initialization of the Gibbs moves. If the active block set is selected as $r = \{1\ldots n-1\}$, the update equation in Eq.(18), will be equal to the one introduced in (MacEachern et al., 1999) as S4 algorithm.

The above schema depends exclusively on local Gibbs moves. As is the case in the application of the Gibbs sampler, we may expect to get stuck in local modes due to slow mixing especially when the posterior distribution is multi modal. In such situations, annealing is a general strategy to pass through low probability barriers. However, as one modifies the target density gradually, finding the correct schedule is crucial. On the other hand, in the SMC framework we don't have to choose a schedule explicitly. We are free to choose any forward kernel, provided we compute the corresponding incremental weight. Here, we propose a forward kernel which targets the modified full conditional distribution, $\pi_n(z_{n,n}, z_{n,r}|z_{n,-u}, \rho_n)$. Note that bridging is achieved simply by changing the novelty parameter of the underlying Dirichlet process to ρ_n. The SMC theory guarantees that we still target the original target density.

A valid backward kernel can be obtained by replacing π_n with the modified version of the target distribution $\pi_n(.|\rho_n)$ in Eq.(12) and the resulting weight update equation can be represented as follows,

$$v^{an}(z_{n-1}, z_n) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{\pi_n(z_{n-1,r}|z_{n-1,-r}, \rho_n)}{\pi_n(z_{n,u}|z_{n,-u}, \rho_n)}. \quad (20)$$

While we are able to escape low probability barriers, the modified full conditional distribution introduces a

further approximation. Thus, we advise still choosing the ρ_n converging to the true κ with the increasing time index n. Note that this is merely a choice, not a requirement in contrast to a tempered Gibbs sampler, where the final density must coincide with the true target.

4.2 Sequential approximation

As we rely on a blocked Gibbs sampler, we are constrained by low dimensional blocks. The key idea in this method is to approximate sequentially to the exact full conditional distributions given by Eq.(14) and Eq.(15). As in the previous section, we are free to choose any approximation to the full conditionals as these are merely used as our proposal density. Asymptotically, the SMC sampler ensures convergence to true target posterior even approximations to these full conditional distributions are defined. Note that the approximations, should be selected as close as possible to the full conditionals to obtain an efficient sampler.

We assume that there are Q clustering variables in the active block and we further enumerate them

$$z_{n,r} = \left\{ z_{n,r_1} \dots z_{n,r_Q} \right\} \tag{21}$$

where r_q denotes the q'th index of the block at time n with $q = 1 \dots Q$. In the sequel, we will design an approximation that enables us to design kernels where the computational load increases linearly with Q. Hence, we can chose an active block with size Q quite large in practice.

We propose the following approximation to the forward kernel K_n,

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1,-u}}(z_{n,-u})\, \hat{\pi}_n(z_{n,u}|z_{n,-u}) \tag{22}$$

where

$$\hat{\pi}_n(z_{n,u}|z_{n,-u}) = \pi_n(z_{n,n}|z_{n,r}, z_{n,-u}, \rho_n) \times \prod_{i=1}^{Q} \pi_{n-1,-r_{\{i+1:Q\}}}(z_{n,r_i}|z_{n,-u}, z_{n,r_{\{1:i-1\}}}, \rho_n). \tag{23}$$

We assume $r_{\{i:j\}}$ is empty for $i > j$. The rationale beyond this choice is as follows: we drop all the observations corresponding to the active block, including the last observation and incorporate them one by one in a new (possibly random) order. Recall that in Eq.(23), $\pi_{n-1,-r_{\{i+1:Q\}}}(z|z', \rho_n) = p(z|z', \{y_{n-1,1:n-1}\} - \{y_{n-1,r_{i+1}} \dots y_{n-1,r_Q}\})$ denotes the modified full conditional distribution given the all observations excluding the ones indexed by $r_{\{i+1:Q\}}$. Note that approximations of this form are quite common in approximate inference for state space models, where q corresponds to a time index; we merely omit the effect of the 'future' observations.

The formulation given by Eq.(23) enables us to recursively calculate and sample the overall kernel density function $\hat{\pi}_n(z_{n,u}|z_{n,-u})$ efficiently with a reasonable complexity even for large values of Q. Note that the scheme introduced in Eq.(23) processes the observations sequentially in the indexed order $\{r_1 \dots r_q\}$ and finally extends the space using the proposal function $\pi_n(z_{n,n}|z_{n,r}, z_{n,-u}, \rho_n)$. Clearly, due to exchangability of the DPM model, there is no need to process the observations in a fixed order. To diminish the effects of the particular processing order, it is preferred to apply a random permutation of the indicies in r at each step of the algorithm.

A similar sequential procedure is also required to approximate the backward kernel given by

$$L_{n-1}(z_n, z_{n-1}) = \delta_{z_{n,-u}}(z_{n-1,-u}) \times \hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r}) \tag{24}$$

where

$$\hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r}) = \prod_{i=1}^{Q} \pi_{n-1,-r_{\{i+1:Q\}}}(z_{n-1,r_i}|z_{n-1,-u}, z_{n-1,r_{\{1:i-1\}}}, \rho'_n). \tag{25}$$

According to the resampling scheme given in Section.4.4 it is convenient to select $\rho'_n = \rho_n$ in order to construct a good approximation to the optimal backward kernel. Note that given the same index order $r_1 \dots r_Q$, Eq.(25) will have the same functional form with the right most hand side of Eq.(23) when $\rho'_n = \rho_n$.

Finally, using the given approximations for the forward and backward kernels the weight update equation can be arranged according to Eq.(9) as follows,

$$v^{sq}(z_{n-1}, z_n) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{\hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r})}{\hat{\pi}_n(z_{n,u}|z_{n,-u})}. \tag{26}$$

4.3 Mixture kernels

In Monte Carlo computations for solving high dimensional complex problems, it is common practice to resort to a collection of kernels rather than committing to a fixed choice. One can define a mixture kernel in the context of a SMC algorithm as follows,

$$K_n(z_{n-1}, z_n) = \sum_{m=1}^{M} \alpha_n^m(z_{n-1}) K_n^m(z_{n-1}, z_n) \tag{27}$$

where $m \in \{1 \dots M\}$ is the mixture label, α_n^m denotes the selection probability of the mixture component at time n, $\sum_{m=1}^{M} \alpha_n^m(z_{n-1}) = 1$, and $K_n^m(z_{n-1}, z_n)$ denotes the forward kernel corresponding to the m'th

component. In order to circumvent the computational burden of Eq.(27), a backward kernel of the form of a mixture is proposed in (DelMoral et al., 2006).

$$L_n(z_n, z_{n-1}) = \sum_{m=1}^{M} \beta_n^m(z_n) L_n^m(z_n, z_{n-1}) \tag{28}$$

where β_n^m is the backward mixture component selection probability at time n, $\sum_{m=1}^{M} \beta_n^m(z_{n-1}) = 1$. Now, this definition enables us to perform importance sampling on an extended space $E \times E \times \mathcal{M}$ by the definition of a latent kernel selector variable $\mathbf{M}_n$, taking values $\mathcal{M} = \{1 \dots M\}$, $m \in \mathcal{M}$. Consequently the weight function for each mixture component can be expressed as given in Eq.(29).

$$v_n(z_{n-1}, z_n, m) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{L_{n-1}^m(z_n, z_{n-1}) \beta_n^m(z_n)}{K_n^m(z_{n-1}, z_n) \alpha_n^m(z_{n-1})} \tag{29}$$

In our work we define three different algorithms labeled as SMC-1, SMC-2 and SMC-3 each utilizes a different kernel. The SMC-1 algorithm employs the forward kernel given by Eq.(18) and updates the weights according to Eq.(19).

The SMC-2 algorithm uses a mixture kernel in order to admit the Gibbs sampler to make global moves in the DPM space. When the selection probabilities α and β are chosen equal and independent of the z_n and z_{n-1} respectively, the mixture weight update function for $m = 1$ and $m = 2$ can be given by Eq.(19) and Eq.(20) respectively.

In SMC-3 we use the mixtures of the forward kernel given by Eq.(18) and Eq.(22) where the sequential construction enables us to define a backward kernel independent from the modified forward kernel parameter ρ_n. The associated weight update functions are calculated according to Eq.(19) and Eq.(26) respectively when selection probabilities α and β are equal and chosen independent of z.

4.4 Algorithmic details

As denoted before, in our work, we propose an active block to be updated as each new observation arrives. In order to limit the computational cost required at each time step we determine a constant block size Q and index the block with $r_1 \dots r_Q$. Similar block update strategies have also been proposed in (Doucet et al., 2006) under the SMC samplers framework. In our scheme the indexes of the active block $r_1 \dots r_Q$ are incremented by Q as each new observation arrives. Whenever the last index value is $r_Q > n$ then we set $r = \{1 \dots Q\}$. Note that according to the sequential construction defined in Eq.(23) the approximations will be less accurate for the algorithm SMC-3 with the increasing size Q.

In order to prevent particle degeneracy, in a SMC framework it is required to perform occasional resampling steps when the effective sample size drops below a predefined threshold. Intuitively, this step selects the high weighted particles and discards the low weighted ones. However, discarding the low weighted particles prematurely may prevent an algorithm to explore promising modes of the time evolving posterior distribution. It is quite common in practice, that a mode initially less dominant becomes more pronounced when a larger fraction of the data is processed. Hence for the DPM model, we found it crucial to apply the resampling step on the modified target distribution $\pi(.|\rho)$ instead of the true target posterior π in order to prevent the low weighted particles to be discarded too early.

We calculate the weights as follows: We first calculate the unnormalized weights for the modified target distribution $\pi_n(.|\rho_n)$ according to $w_n^{'} = w_n \times \gamma_n(z_n|\rho_n)/\gamma_n(z_n)$. Assuming that, $\{W_n^{'(i)}\}$ represents the normalized weights, we apply systematic resampling if effective sample size, $N_{eff} = 1/\sum_{i=1}^{N_p} (W_n^{'(i)})^2$ is below a predefined threshold. Following the resampling, a reweighting step, $w_n = \gamma_n(z_n)/\gamma_n(z_n|\rho_n)$, is being carried out, in order to find the weights approximating to the true target posterior.

5 Test Results

Our goal in this section is to illustrate the effectiveness of the SMC samplers framework for online inference in DPM models. For this purpose, we compare performance of SMC samplers each detailed in Section.4.3, namely; SMC-1, SMC-2, SMC-3, Particle filter (PF) (MacEachern et al., 1999; Fearnhead, 2004) and a batch algorithm, Gibbs sampler (GS) (MacEachern, 1994). Performance has been reported in terms of log-marginal likelihoods, mean, variance estimates and respective standard errors. Mixture density estimates are also provided for visual comparison.

The problem is the standard Gaussian mixture density estimation problem with unknown number of components. Our model is standard and assumes that observations y are drawn from a univariate Gaussian with unknown mean μ and variance σ^2, $\theta = \{\mu, \sigma^2\}$ where the number of mixtures are unknown. The distribution of the parameters μ and σ^2 are chosen as normal and inverse-gamma, respectively to ensure the conjugacy condition.

Table 1: True model parameters

	Mixture weights	Mean	Std. dev.
Data-1 (D-1)	1/3,1/3,1/3	0,1.5,3	0.5,0.5,0.5
Data-2 (D-2)	1/2,1/6,1/3	0,2,4	0.5,0.5,2.5

Apart from the resampling threshold and the number of particles, several algorithm parameters need to be set: The selection probabilities of the forward and the backward kernels (α and β), active block size (q), and the parameter sequence ρ_n. The selection probabilities determine the shape of the forward and backward kernels therefore an appropriate choice is crucial. Selection probabilities of the forward and backward kernel are set to $\alpha^m = \{0.9, 0.1\}$ and $\beta^m = \{0.9, 0.1\}$ respectively. Note that $m = 2$ corresponds to the modified kernel component and we practically observed that a small weight is often enough to obtain a good mixing property. Increasing the weighting of the modified kernel often increase the algorithms ability to explore new modes. Even a single kernel where $\alpha^m = \{0, 1\}$ can be used for certain dataset where modes are highly isolated. The parameter sequence ρ_n is updated according to a geometric update function (Neal, 2001) where the common parameter is set to $1/150$ and the initial value is set to $\rho_1 = 1$. The active block size q is set to 4. This choice seems to balance well computational burden with inference quality.

To alleviate the degeneracy, we applied systematic resampling scheme. The resampling scheme for SMC-2 and SMC-3 is applied according to Section 4.4. For a fair comparison the particle size is selected as $N_p = 1000$ for particle filter, $N_p = 200$ for SMC algorithms and we performed 1000 iterations by Gibbs sampler where the first 300 were used for the burn-in period. All the results are reported for 100 independent Monte Carlo runs. We selected two test sets (D-1 and D-2) generated from a Gaussian mixture model. Each data set has 1000 points, and the results are reported sequentially for 200, 500 and 1000 observations. Both datasets are generated from a model comprising of three mixture components with parameters given in Table.1. In order to evaluate the performance of the proposed kernels we performed two tests that aim to assess the mixing property (ability to escape local modes) as well as the consistency and quality of the estimate (bias and low variance).

Table 2: Estimated average Log-marginal likelihoods, mean values and respective Monte Carlo errors (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS

Dataset-1 (D-1), $\kappa = 0.05$				
		Estimated Mean		
Algo.	Log-marg.	200	500	1000
SMC-1	-723.4 (102.8)	2.11 (0.014)	2.51 (0.233)	2.67 (0.243)
SMC-2	-711.2 (4.41)	2.15 (0.006)	3.10 (0.025)	3.10 (0.020)
SMC-3	-711.1 (3.22)	2.15 (0.007)	3.09 (0.011)	3.09 (0.010)
PF	-727.6 (52.9)	2.10 (0.015)	2.35 (0.181)	2.49 (0.249)
GS				2.69 (0.239)

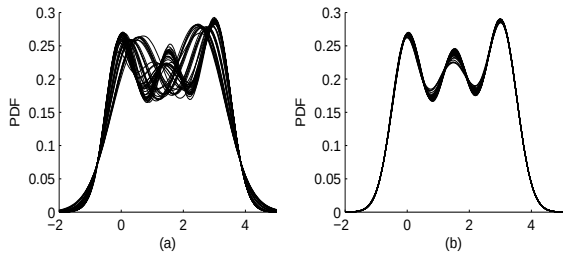

Figure 1: Estimated mixture densities by the (a) SMC-1 and (b) SMC-3 algorithm for 50 Monte Carlo runs.

Sampling based inference schemes get stuck in local modes of the posterior distribution, particularly when the novelty parameter is chosen too small for the given problem. In order to compare the mixing property of the proposed algorithms we set the novelty parameter to a very low value of $\kappa = 0.05$ and compare the mixture densities estimated by SMC-1 and SMC-3 algorithms, respectively. We performed the test by generating a total of 1000 observations from the model D-1 which comprise three overlapping mixture components. As a gold standard reference we performed a very long Gibbs sampler run and observed that the estimated number of components is 2.16, 3.09 and 3.11 for $200, 500$ and 1000 observations consecutively. Fig.1 (a) and (b), respectively illustrate the estimated mixture densities obtained by SMC-1 and SMC-3 for 50 Monte Carlo runs. It is clear that SMC-3 can represent all tree components of the mixture density for all runs. However, in nearly half of the runs, the SMC-1 estimates 2 mixture components because it gets stuck to a local mode. The mean estimates of log-marginal likelihood and the number of components are given in Table.2 for SMC-1, SMC-2, SMC-3, PF and GS. The results show that the mean estimate of the SMC-2 and SMC-3 are very close to the long run estimate of the Gibbs sampler, however SMC-1, PF and GS underestimate the mean value even when the observation size is 1000 and GS requires a longer burn-in period in order to converge to the true posterior distribution. SMC-2 and SMC-3 are also superior in means of marginal log-likelihoods.

In order to measure the consistency of the proposed algorithms, the performance of the algorithms for different parameter settings are reported in Table.3 for D-1 and D-2, respectively. The novelty parameter is set to $\kappa = 0.5$ which avoids the algorithms to stuck at a local solution. The mean estimate for the long Gibbs sampler run is 3.73 for D-1 and 4.63 for D-2 at $n = 1000$. As it is shown in Table.3, PF, GS and SMC algorithms provide very close mean estimates to the long run Gibbs sampler for D-1. Monte Carlo standard error of the mean estimate for particle filter grad-

Table 3: Estimated average Log-marginal likelihoods, mean values and respective Monte Carlo errors (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS

Dataset-1 (D-1), $\kappa = 0.5$				
		Estimated mean		
Algo.	Log-marg.	200	500	1000
SMC-1	-708.9 (1.54)	2.99 (0.038)	3.61 (0.061)	3.71 (0.056)
SMC-2	-708.6 (0.96)	3.00 (0.025)	3.61 (0.044)	3.69 (0.036)
SMC-3	-708.9 (1.20)	2.96 (0.025)	3.60 (0.042)	3.69 (0.038)
PF	-712.4 (9.86)	2.98 (0.041)	3.70 (0.272)	3.79 (0.293)
GS				3.68 (0.055)
Dataset-2 (D-2), $\kappa = 0.5$				
		Estimated mean		
Algo.	Log-marg.	200	500	1000
SMC-1	-1117.3 (0.35)	4.14 (0.035)	4.54 (0.068)	4.65 (0.091)
SMC-2	-1117.3 (0.31)	4.14 (0.021)	4.53 (0.064)	4.63 (0.129)
SMC-3	-1117.2 (0.29)	4.13 (0.019)	4.50 (0.054)	4.58 (0.086)
PF	-1117.7 (0.98)	4.14 (0.030)	4.56 (0.119)	4.73 (0.281)
GS				4.68 (0.095)

ually increases with the observation size and reaches to a value of 0.293 at $n = 1000$ whereas all three SMC samplers achieve approximately 8 times lower errors. It can be concluded that all SMC algorithms provide a significant performance improvement over PF with the same computational cost and they are also more reliable. When we compare the SMC-1, SMC-2 and SMC-3, non of the algorithms outperform the others in means of Monte Carlo error. The results also show that performance of the SMC algorithms and GS are comparable for the dataset D-1 when $\kappa = 0.5$.

A similar performance has also been reported for the dataset D-2. The mean estimate of PF, GS, SMC and the long run Gibbs sampler are very close. All three SMC algorithms and GS provide comparable Monte Carlo errors for mean estimates while they are lower for PF. The results obtained for D-1 and D-2 also indicate that, SMC-2 and SMC-3 achieve reliable estimates at different parameter sets.

6 Conclusion

In this paper we propose novel sequential Monte Carlo algorithms for the DPM model with a conjugate prior. In contrast to the existing sequential importance sampling methods, the local moves are designed to update clustering labels that enable the introduced algorithms to obtain efficient samples from the time evolving posterior even for large dataset sizes.

We have evaluated the performance of the conventional particle filter, Gibbs sampler and SMC samplers with different kernel setting, on two different datasets. Test results show that SMC based methods provide more reliable estimates compared to conventional particle filter and proposed mixture kernel can better represent the modes of the posterior distribution compared to a SMC sampler utilizing Gibbs moves. We also conclude that the SMC framework is a competitive alternative to the conventional Gibbs sampler for the DPM models. As future work, we envision various applications in hierarchical Bayesian models with a DPM prior. Finally, while we have concentrated exclusively on the conjugate setting, we believe that the actual added benefit of the SMC framework can be realized in the non-conjugate setting where the model parameters need also be sampled.

References

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

Blei, D. M. and Jordan, M. I. (2004). Variational methods for the dirichlet process,. In *Proceedings of the 21st International Conference on Machine Learning*.

Chen, R. and Liu, J. S. (2000). Mixture kalman filters. *J. Roy. Stat. Soc. B Stat. Meth.*, 62:493–508.

DelMoral, Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *J. Roy. Stat. Soc. B Stat. Meth.*, 63:11–436.

Doucet, A., Briers, M., and Senecal, S. (2006). Efficient block sampling strategies for sequential monte carlo methods. *J. Comput. Graph. Stat.*, 15(3):693–711.

Escobar, M. and West, M. (1992). Computing bayesian nonparametric hierarchical models. Technical report, Duke University, Durham, USA.

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

Jain, S. and Neal, R. (2000). A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *J. Comput. Graph. Stat.*, 13:158–182.

Liu, J. S. (1996). Nonparametric hierarchal bayes via sequential imputations. *Ann. Statist.*, 24:910–930.

MacEachern, S. N. (1994). Estimating normal means with a conjugate style dirichlet process prior. *Comm. Statist. Simulation Comput.*, 23:727–741.

MacEachern, S. N., Clyde, M., and Liu, J. (1999). Sequential importance sampling for nonparametric bayes models: the next generation. *Can. J. Stat.*, 27:251–267.

Neal, R. (2001). Annealed importance sampling. *Statist. Comput*, 11:125–139.

Quintana, F. A. (1996). Nonparametric bayesian analysis for assessing homogeneity in k×l contingency tables with fixed right margin totals. *J. Am. Stat. Assoc.*, 93:1140–1149.

Quintana, F. A. and Newton, M. A. (1998). Computational aspects of nonparametric bayesian analysis with applications to the modeling of multiple binary sequences. *J. Comput. Graph. Stat.*, 9:711–737.