

# Annealed SMC Samplers for Nonparametric Bayesian Mixture Models

Yener Ulker, Bilge Günsel, *Member, IEEE*, and A. Taylan Cemgil, *Member, IEEE*

**Abstract**—We develop a novel online algorithm for posterior inference in Dirichlet Process Mixtures (DPM). Our method is based on the Sequential Monte Carlo (SMC) samplers framework that generalizes sequential importance sampling approaches. Unlike the existing methods, the framework enables us to retrospectively update long trajectories in the light of recent observations and this leads to sophisticated clustering update schemes and annealing strategies that seem to prevent the algorithm to get stuck around a local mode. The performance has been evaluated on a Bayesian Gaussian density estimation problem with an unknown number of mixture components. Our simulations suggest that the proposed annealing strategy outperforms conventional samplers. It also provides significantly smaller Monte Carlo standard error with respect to particle filtering given comparable computational resources.

**Index Terms**—Dirichlet process mixtures, particle filtering, sequential Monte Carlo methods.

## I. INTRODUCTION

IN recent years, there has been a surge of interest in Bayesian nonparametric methods in machine learning [1] and signal processing [2]. Here, researchers use highly structured and adaptive models where the model order is to be determined automatically by the data. The Dirichlet Process Mixtures (DPM) is the key building block in such models for a broad range of applications.

Unfortunately, exact computation for the DPM model is intractable. The development of efficient and accurate approximate inference strategies for the DPM are crucial so that such models can be used effectively in practical applications.

Particle filtering is a popular method for inference in DPM models [3], [4]. Existing particle filtering based approaches update the representation of the posterior distribution sequentially as each new observation arrives. Especially for large datasets, sequential processing is preferable over batch approaches (such as [5]). Advantages include reduction in computational cost and storage requirement and a natural tempering effect that helps the inference. However, a problem with sequential inference is the potential danger of accumulation of Monte Carlo errors and

consequent decrease in the estimation quality. In this work we propose a novel sequential Monte Carlo sampler for inference in the DPM model. The proposed method is an annealed sequential Monte Carlo (SMC) sampler that enables us to design sophisticated clustering update schemes and ensures convergence to the true target distribution asymptotically. An important feature of our approach is that the proposed algorithm updates past particle trajectories in light of recent observations to avoid error accumulation. This approach is not feasible in the particle filtering setting.

The performance of the algorithm is evaluated in a Bayesian density estimation problem with unknown number of components where the prior on parameters is conjugate. It is shown that the proposed annealing scheme accurately represents the target posterior and it provides smaller standard error compared to particle filtering. In contrast to our previous work [6] which utilize mixture kernels in order to target the true DPM posterior, here we focus on a single proposal kernel targeting an annealed posterior distribution and update the weights according to the true target density. The conference version of this work is presented in [7].

## II. DIRICHLET PROCESS MIXTURES (DPM)

In this section we will define the Dirichlet process mixtures (DPM) model. We will focus on the infinite mixture interpretation where observations arrive sequentially. It is known that due to exchangeability and the specific construction of this model, the actual order is irrelevant [8]. We refer to the time index as  $n$  and the observation sequence obtained until time  $n$  is denoted 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 corresponding cluster label and,  $k_n \in \{1 \dots n\}$  represents the number of existing 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 i.i.d. and the observations are independent of each other conditional on the cluster variable  $z_{n,i}$ . Hence the DPM defines the joint density  $\pi(z_n, \theta_n)$  that can be expressed as

$$\pi(z_n, \theta_n) \propto p(z_n) \prod_{j=1}^{k_n} p(\theta_{n,j}) \prod_{i=1}^n p(y_{n,i} | \theta_{n,z_{n,i}}). \quad (1)$$

The prior on clustering variable vector  $z_n$  is formulated by (2) in a recursive way,

$$p(z_{n,i+1} = j | z_{n,\{1:i\}}) = \begin{cases} \frac{l_j}{i+\kappa}, & j = 1, \dots, k_i \\ \frac{\kappa}{i+\kappa}, & j = k_i + 1 \end{cases} \quad (2)$$

where  $k_i$  is the number of clusters in the assignment  $z_{n,\{1:i\}}$ . In (2),  $l_j$  is the number of observations that  $z_{n,\{1:i\}}$  assigns to

Manuscript received April 26, 2010; revised August 11, 2010; accepted August 13, 2010. Date of publication September 02, 2010; date of current version November 15, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thia Kirubarajan.

Y. Ulker is with the Department of Electronics and Communication Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey (e-mail: yeneru@yahoo.com; yenerulker@itu.edu.tr).

B. Günsel is with the Multimedia Signal Processing and Pattern Recognition Laboratory, Department of Electronics and Communications Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey (e-mail: gunselb@itu.edu.tr).

A. T. Cemgil is with the Department of Computer Engineering, Bogazici University, 34342 Bebek, Istanbul, Turkey (e-mail: taylan.cemgil@boun.edu.tr).

Digital Object Identifier 10.1109/LSP.2010.2072919

cluster  $j$  and  $\kappa$  is the positive valued 'novelty' parameter [8]. In our model we assume that the parameters  $\theta_n$  satisfies the conjugacy condition hence our aim is to estimate the sequentially evolving posterior density of the DPM model that is intractable due to the exploding cardinality of  $z_n$ .

### III. SEQUENTIAL MONTE CARLO SAMPLERS

In sequential inference, one samples from a sequence of target densities,  $\pi_1(x_1) \cdots \pi_n(x_n)$ , evolving with  $n$  each defined on a measurable space  $(E_n, \mathcal{E}_n)$  where  $x_n \in E_n$ . Building on the work of Neal [5], Del Moral *et al.* [9] proposed an auxiliary variable technique which solves the sequential importance sampling problem in an extended space  $E^n = \{E_1 \times \cdots \times E_n\}$ . SMC sampler performs importance sampling between the joint importance distribution  $\eta_n(x_{1:n})$  and the artificial joint target distribution defined by  $\tilde{\pi}_n(x_{1:n}) = \tilde{\gamma}_n(x_{1:n})/Z_n$  where  $Z_n$  denotes the normalizing constant and

$$\tilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n) \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k). \quad (3)$$

$L_n$  is a backward Markov Kernel from space  $E_{n+1}$  to  $E_n$ . Note that,  $\tilde{\pi}_n(x_{1:n})$  admits  $\pi_n(x_n) = \gamma_n(x_n)/Z_n$  as a marginal by construction therefore the resultant weight function ensures convergence to the true target density.

Assume that a set of weighted particles  $\{W_{n-1}^i, X_{1:n-1}^i\}_{i=1}^{N_p}$  approximate  $\tilde{\pi}_{n-1}$  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,  $\tilde{\gamma}_n(x_{1:n})/\eta_n(x_{1:n})$ , associated with the extended particles are calculated according to  $w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})v_n(x_{n-1}, x_n)$  where the incremental term of weight equation,  $v_n(x_{n-1}, x_n)$  is

$$v_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}. \quad (4)$$

The design of efficient sampling schemata hinges on properly choosing the backward kernel  $L_{n-1}$ . Assuming  $K_n$  is an Monte Carlo Markov Chain (MCMC) kernel of invariant distribution  $\pi_n$ , an approximate backward kernel can be formulated as shown in (5):

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \quad (5)$$

which is accepted as a good approximation for  $\pi_{n-1} \approx \pi_n$  and yields to the incremental weight  $v_n(x_{n-1}, x_n) = \gamma_n(x_{n-1})/\gamma_{n-1}(x_{n-1})$ .

### IV. SMC SAMPLERS FOR DPM MODELS

Let  $K_n$  denote the forward kernel that will be used to generate samples from the posterior distribution formulated in (1). We first partition an assignment vector  $z_n = \{z_{n,r}, z_{n,d}, z_{n,n}\}$  where  $r$  is a subset of  $\{1, \dots, n-1\}$ , a set of not necessarily consecutive indices, and  $d = \{1, \dots, n-1\} - r$ . Throughout the text we will call the set  $z_{n,r}$  as the active block. We define  $u = r \cup \{n\}$ , and denote  $-u \equiv d$ .

Let us define the forward kernel as follows:

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_{n,-u}) K_n(z_{n,n}, z_{n,r}|z_{n-1}) \quad (6)$$

where  $K_n(z_{n,n}, z_{n,r}|z_{n-1})$  is a valid MCMC kernel applying a single Gibbs iteration targeting the full conditional distribution  $\pi_n(z_{n,n}, z_{n,r}|z_{n,-u})$ .

The corresponding backward kernel can be obtained by substituting (6) into (5) that yields the incremental weight update equation

$$v_n(z_{n-1}, z_n) = \frac{\gamma_n(z_{n-1}, z_n, -u)}{\gamma_{n-1}(z_{n-1})}. \quad (7)$$

Note that as a consequence of using the MCMC kernel  $K_n$ , (7) is independent from the kernel initialization. When the active block set is selected as  $r = \{1 \dots n-1\}$ , we obtain the update rule (7) introduced in [10] as S4 algorithm. Intuitively, the MCMC kernel updates the active block using a Gibbs sampler and constructs the proposal distribution using the sequence of full conditional distributions.

In a sequential problem the posterior distribution changes over time and new modes of the posterior distribution may emerge as new observations are received. The algorithm must have a good mixing property to explore the modes of the time evolving posterior distribution and to achieve a good approximation to the true target posterior. However, conventional sequential and batch algorithms based on the Gibbs sampler may fail to represent the modes of the true target posterior due to the slow convergence property of the Gibbs samplers. This is particularly when the posterior distribution has a multi modal form where the modes are isolated [5]. To deal with this problem, in the next subsection we introduce an algorithm that converges to the true DPM posterior as the new observations are received sequentially.

#### A. Annealing Procedure

The conventional approach presented in Section IV applies Gibbs moves to each particle in order to obtain weighted samples from a sequence of target distributions denoted as  $\pi_1(z_1), \dots, \pi_n(z_n)$ . This paper proposes an annealing scheme to improve the efficiency of posterior estimation. In the literature annealing schemes have been widely used to handle isolated modes in batch processing. It is adopted to importance sampling to construct the proposal distribution suitable to sampling of the true target distribution [5].

To achieve our goal let us construct an annealed time evolving target posterior as  $\pi'_1(z_1), \dots, \pi'_n(z_n)$ ,  $k = \{1 \dots n\}$ , where  $\pi'_k$  is the annealed target posterior defined as

$$\pi'_k(z_k) = \pi_k(z_k|\kappa = \alpha_k). \quad (8)$$

Annealing is achieved by changing the novelty parameter of the underlying Dirichlet process which is set to  $\alpha_k$  in (8). Note that  $\alpha_k$  is a parameter of the prior distribution of number of components where a higher value yields higher number of mixtures. The idea behind constructing a sequence of annealed target posterior distribution is to obtain a class of intermediate distributions by selecting a  $\alpha_k$  value which is higher than the true model novelty parameter  $\kappa$  and provide a well defined support to the time evolving target posterior. In other words, the annealed distributions can be interpreted as an underlying DPM model of which the parameters are relaxed in order to obtain an annealed posterior which is easy to sample.

In order to sample the sequence of annealed target distributions, let us define a forward kernel as follows:

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_n, -u) K_n(z_{n,n}, z_{n,r} | z_{n-1}) \quad (9)$$

where  $K_n(z_{n,n}, z_{n,r} | z_{n-1})$  is an MCMC kernel which targets the conditional distribution  $\pi'_n(z_{n,n}, z_{n,r} | z_{n,-u})$ . Using (5), the backward kernel can be written as in (10):

$$L_{n-1}(z_n, z_{n-1}) = \frac{\pi'_n(z_{n-1}) K_n(z_{n-1}, z_n)}{\pi'_n(z_n)} \quad (10)$$

and the incremental weights for the annealed target posterior can be obtained as follows:

$$v'_n(z_{n-1}, z_n) = \frac{\gamma'_n(z_n) \pi'_n(z_{n-1}, r | z_{n-1}, -u)}{\gamma'_{n-1}(z_{n-1}) \pi'_n(z_n, z_{n,r} | z_{n,-u})} \quad (11)$$

where  $\pi'_n(z_n) = \gamma'_n(z_n) / Z_n$  and the weights associated with the particles can be calculated according to  $w'_n(z_{1:n}) = w'_{n-1}(z_{1:n-1}) \times v'_n(z_{n-1}, z_n)$ . Assuming  $\{W_n^{(i)}\}$  represents the normalized weights approximating to  $\pi'_n(z_n)$ , we perform a resampling step if effective sample size,  $N_{eff} = 1 / \sum_{i=1}^{N_p} (W_n^{(i)})^2$ , is below a predefined threshold. Finally, in order to approximate the target distribution  $\pi_n(z_n)$ , we reweight the particles according to  $w_n(z_{1:n}) = w'_n(z_{1:n}) \times v_n(z_n)$  where  $v_n(z_n) = \gamma_n(z_n) / \gamma'_n(z_n)$ .

Specification of the active block size  $r$  shown in (11) is an important issue in the design of the proposed sampler. In order to limit the computational cost required at each time step we initially determine a constant block size  $Q$  and index the block with  $r_1 \dots r_Q$ . The indexes of the active block is incremented by  $Q$  as each new observation is received. The blocks do not overlap to each other and update scheme is cycled whenever all the clustering labels up to time  $n$  are updated. Note that similar block update strategies are also used in [11] under the SMC samplers framework.

### B. The Annealing Parameter

As denoted above the sequence of annealed posterior distributions,  $\pi'_1(z_1), \dots, \pi'_n(z_n)$  is constructed by updating the annealing parameter  $\alpha_n$  of the underlying DPM model shown in (8). At each time step of the algorithm  $\alpha_n$  is updated according to a geometric spacing function

$$\alpha_n = \alpha_{n-1} + c_\alpha(\kappa - \alpha_{n-1}) \quad (12)$$

where  $\alpha_1 > 0$ ,  $\alpha_{n-1} > \alpha_n$  and  $c_\alpha$  is the common parameter that determines the amount of spacing at each time step. Note that in [5] it is reported that to change the annealing parameter according to geometric spacing of  $\alpha_k$  is suitable when the  $\pi'_n$  varies smoothly with time.

In our framework, we construct the sequence of annealed distributions by setting an initial value  $\alpha_1$  and updating  $\alpha_n$  as each new observation arrives. Intuitively the initial value for  $\alpha_1$  and the common parameter  $c_\alpha$  are set empirically in order to form the intermediate distributions that are not too far apart from the true target density  $\pi_n$ . We note that, in conventional annealing approaches, where one modifies the target density gradually, finding the correct schedule is a hard but crucial task. In contrast, in the SMC framework we don't have to choose a schedule

very strictly. We are free to choose any forward kernel, provided we compute the corresponding incremental weight—we will be sampling from the correct target at any given time.

## V. TEST RESULTS AND CONCLUSIONS

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 three samplers namely; the SMC-G which utilizes conventional Gibbs moves on the DPM space [10], the proposed SMC sampler (SMC-A), the SMC-M algorithm that utilize a mixture of Gibbs moves and approximate Gibbs moves based on sequential approximation [6] and the Particle filter (PF) [4]. Performance has been reported in terms of log-marginal likelihoods, mean estimates and respective standard errors. Mixture density estimates are also provided for visual comparison.

Algorithms are evaluated on a infinite Gaussian mixture density estimation problem. Observations are drawn from a univariate Gaussian with  $\theta = \{\mu, \sigma^2\}$  where  $\mu$  is the mean and  $\sigma^2$  is the variance. The conjugate prior distributions are chosen as normal and inverse-gamma, respectively.

To alleviate the degeneracy, a systematic resampling scheme is applied for sequential algorithms when  $N_{eff} < 3/4 N_p$ . For a fair comparison the number of particles is selected as  $N_p = 1000$  for PF,  $N_p = 100$  for SMC-A algorithm and  $N_p = 200$  for the SMC-M algorithm where the active block size  $Q$  is set to 9 and 4 for the SMC-A and SMC-M algorithms respectively. Note that block size determines the approximation introduced by the kernels for the SMC-M algorithm [6]. The results are reported for 100 independent Monte Carlo runs for each model. The initial annealing parameter for annealed target distribution is set to  $\alpha_1 = 1$  and it is geometrically updated according to (12) at each time step where the common parameter,  $c_\alpha$ , is set to  $1/100$ .

Two test sets (D-1 and D-2) are generated from a Gaussian mixture model comprising three mixture components with parameters given in Table I where  $\mu_i$ ,  $\sigma_i$ , and  $p_i$ , for  $i \in \{1 \dots 3\}$ , denote the mean, standard deviation and the mixture weight for each component, respectively. In order to evaluate the performance on real data, we also performed the tests on the speech data set (D-3) publicly available at [12]. Reported results are obtained for the emotional state “sad” where the actual number of mixture components is priorly unknown. Each test set has a total of 1000 points and the results are reported sequentially for 200, 500, and 1000 samples.

In order to illustrate the mixing capability of the proposed algorithm we set the novelty parameter to a very low value of  $\kappa = 0.05$ . Note that a low  $\kappa$  will probably cause the posterior to have isolated modes hence this test aims to assess the mixing property (ability to escape local modes) of the algorithms. 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. In Fig. 1, the mixture densities are plotted for each run of the PF, SMC-G, SMC-A and SMC-M algorithms, respectively. It is clear that SMC-A and SMC-M can represent all three components of the mixture density in all runs of the algorithms whereas SMC-G and PF commonly gets stuck at a local mode and fits two mixture components to the data for

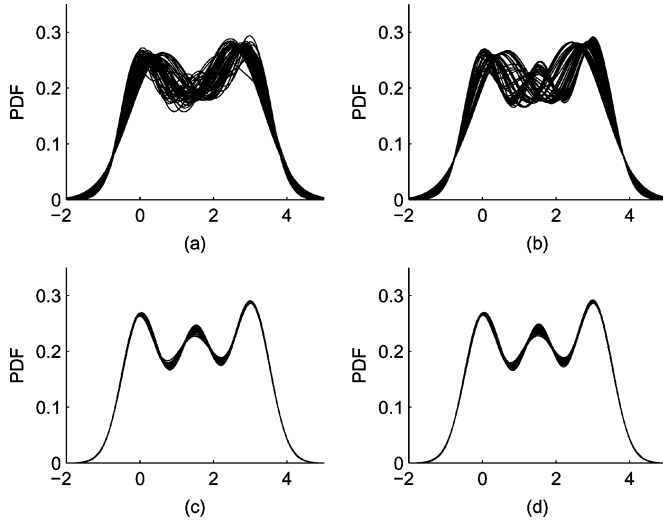


Fig. 1. Estimated mixture densities by the (a) PF, (b) SMC-G, (c) SMC-A, and (d) SMC-M algorithm for 50 Monte Carlo runs. SMC-A and SMC-M represent all tree components of the mixture density in all runs.

TABLE I  
TRUE MODEL PARAMETERS

	$p_1, p_2, p_3$	$\mu_1, \mu_2, \mu_3$	$\sigma_1, \sigma_2, \sigma_3$
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

several runs (more than the half) of the algorithm. We also reported the log-marginal likelihood, mean estimate of the number of components and respective standard errors (in parenthesis) in Table II for SMC-G, SMC-A, SMC-M, and PF. The results illustrate that SMC-A and SMC-M are able to converge to the three components for a small number of observations, however the SMC-G and PF algorithms do not converge to the true posterior even when the observation size is 1000. It is also clear that SMC-A has much lower standard error compared to SMC-G and PF in means of log-marginal likelihoods and the mean estimates whereas a slight improvement is achieved over SMC-M.

In order to examine dependency of the performance of the algorithms on different datasets and parameter settings, we set the novelty parameter to  $\kappa = 0.5$  and report the results in Table II for dataset D-2. It is clear that PF and SMC algorithms provide very close mean estimates. However, SMC-G, SMC-A, and SMC-M can achieve significantly lower standard error compared to PF at  $n = 1000$ . This result shows that SMC algorithms are more reliable with the same computational cost. Moreover SMC-A achieves comparable performance to SMC-G and SMC-M in means of standard error when  $\kappa = 0.5$  while it provides similar mean estimates.

Finally we compared the performance for dataset D-3, where the novelty, initial annealing and the common parameter are set to  $\kappa = 0.05$ ,  $\alpha_1 = 0.25$  and  $c_\alpha = 1/2000$  respectively. As a gold standard reference the results of a very long Gibbs sampler run are found as 2.53, 3.35, and 4.10 for 200, 500, and 1000 observations consecutively. The results given in Table II shows that the SMC-A and SMC-M provides closer estimates to the long Gibbs sampler run particularly when  $n = 1000$  whereas SMC-G and PF underestimates the mean value. Similarly SMC-A outperforms SMC-G and PF in means of log mar-

TABLE II  
ESTIMATED AVERAGE LOG-MARGINAL LIKELIHOODS, MEAN VALUES AND RESPECTIVE MONTE CARLO STANDARD ERRORS (IN PARENTHESIS)

D-1						
Algo.	$\kappa$	Log-marg.	Mean Estimate			
			200	500	1000	
SMC-G	0.05	-723.4 (10.1)	2.11 (0.118)	2.51 (0.152)	2.67 (0.493)	
SMC-A	0.05	-710.8 (1.61)	2.15 (0.070)	3.07 (0.077)	3.07 (0.071)	
SMC-M	0.05	-711.1 (1.78)	2.15 (0.083)	3.09 (0.104)	3.09 (0.115)	
PF	0.05	-727.6 (7.27)	2.10 (0.122)	2.35 (0.425)	2.49 (0.491)	
D-2						
SMC-G	0.5	-1117.3 (0.59)	4.14 (0.187)	4.54 (0.260)	4.65 (0.266)	
SMC-A	0.5	-1117.3 (0.52)	4.14 (0.158)	4.53 (0.244)	4.63 (0.330)	
SMC-M	0.5	-1117.2 (0.53)	4.13 (0.137)	4.50 (0.232)	4.58 (0.293)	
PF	0.5	-1117.7 (0.99)	4.14 (0.173)	4.56 (0.345)	4.73 (0.530)	
D-3						
SMC-G	0.05	-2052.1 (2.26)	2.50 (0.197)	3.09 (0.447)	3.58 (0.452)	
SMC-A	0.05	-2050.6 (0.30)	2.61 (0.424)	3.40 (0.378)	4.04 (0.320)	
SMC-M	0.05	-2051.1 (1.39)	2.60 (0.303)	3.35 (0.360)	4.01 (0.401)	
PF	0.05	-2052.8 (2.54)	2.48 (0.251)	3.06 (0.500)	3.39 (0.573)	

ginal likelihood and achieves lower standard error compared to the SMC-M algorithm.

We have developed a novel online algorithm based on the SMC samplers framework [9]. The key idea of our method is maintaining an intermediate (annealed) distribution as a surrogate target for the SMC algorithm where resampling is carried out according to this annealed distribution. Consequently, we use this surrogate density as a proposal to the true target where we can calculate the correct weights without any extra computational cost. Intuitively, we are using the SMC machinery to compute a good proposal density. This strategy enables us to maintain a diverse particle set that seems to be crucial in obtaining an efficient sampler. The test results show that proposed algorithm is much more efficient compared to conventional methods particularly when DPM target posterior distribution has isolated modes.

## REFERENCES

- [1] D. Blei and M. Jordan, "Variational inference for Dirichlet process mixtures," *J. Bayesian Anal.*, vol. 1, pp. 121–144, 2006.
- [2] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for linear dynamic models with Dirichlet process mixtures," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 71–84, Jan. 2008.
- [3] F. A. Quintana, "Nonparametric Bayesian analysis for assessing homogeneity in  $k \times 1$  contingency tables with fixed right margin totals," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1140–1149, 1996.
- [4] P. Fearnhead, "Particle filters for mixture models with an unknown number of components," *J. Statist. Comput.*, vol. 14, pp. 11–21, 2004.
- [5] R. Neal, "Annealed importance sampling," *Statist. Comput.*, vol. 11, pp. 125–139, 2001.
- [6] Y. Ulker, B. Gunsul, and A. T. Cemgil, "SMC samplers for Dirichlet process mixtures," in *Proc. Thirteenth Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2010.
- [7] Y. Ulker, B. Gunsul, and A. T. Cemgil, "Annealed SMC samplers for Dirichlet process mixture models," in *Proc. Twentieth Int. Conf. Pattern Recognition (ICPR)*, 2010, accepted for publication.
- [8] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statist. Sin.*, vol. 4, pp. 639–650, 1994.
- [9] D. Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *J. Roy. Statist. Soc. B, Statist. Meth.*, vol. 63, pp. 11–436, 2006.
- [10] S. N. MacEachern, M. Clyde, and J. Liu, "Sequential importance sampling for nonparametric bayes models: The next generation," *Can. J. Statist.*, vol. 27, pp. 251–267, 1999.
- [11] A. Doucet, M. Briers, and S. Senecal, "Efficient block sampling strategies for sequential Monte Carlo methods," *J. Comput. Graph. Statist.*, vol. 15, pp. 693–711, 2006.
- [12] [Online]. Available: <http://pascal.kgw.tu-berlin.de/emodh>