

ON THE SEGMENTATION OF SWITCHING AUTOREGRESSIVE PROCESSES BY NONPARAMETRIC BAYESIAN METHODS

Shishir Dash and Petar M. Djurić

Department of Electrical and Computer Engineering,
Stony Brook University, Stony Brook, NY 11794

ABSTRACT

We demonstrate the use of a variant of the nonparametric Bayesian (NPB) forward-backward (FB) method for sampling state sequences of hidden Markov models (HMMs), when the continuous-valued observations follow autoregressive (AR) processes. The goal is to get an accurate representation of the posterior probability of the state-sequence configuration. The advantage of using NPB samplers towards this end is well-known; one need not specify (or heuristically estimate) the number of states present in the model. Instead one uses hierarchical Dirichlet processes (HDPs) as priors for the state-transition probabilities to account for a potentially infinite number of states. The FB algorithm is known to increase the mixing rate of such samplers (compared to direct Gibbs), but can still yield significant spread in segmentation error. We show that by approximately integrating out some parameters of the model, one can alleviate this problem considerably.

Index Terms— hidden Markov model, autoregressive process, segmentation, hierarchical Dirichlet process, Gibbs sampling, nonparametric Bayesian

1. INTRODUCTION

In many research problems, automatic segmentation of continuous-valued time series data is a primary goal [1]. Some well-known example problems are: (a) detecting the order of nucleotides in a DNA sequence [2], (b) finding the sequence of speech units (words or phonemes) from a particular speech signal [3], (c) detecting the sequence of human motions from measurements of positions, velocities or other data [4], and (d) finding the effects of major world events on stock-market returns [5]. A common approach to such problems is through the theory of hidden Markov models (HMMs) in which the parameters of the observation probability distribution function (pdf) are assumed to be dependent only on the value of the hidden state, and the hidden state sequence is assumed to be a Markov chain [3]. For instance, in a first-order HMM, the state-evolution is governed by the pdf $P(s_j = k | s_{j-1} = l, \psi) = \psi_{l,k}$, where s_j denotes the state at time-point j , ψ denotes the transition probability matrix, and the element in ψ 's l -th row and k -th column, $\psi_{l,k}$ denotes the probability of transitioning from state l to state k . The observations x_j are assumed to be governed by $P(x_j | s_j = k, \theta) = L(x_j; \theta_k)$, where θ denotes the set of likelihood parameters for state k . Thus, the observations are conditionally independent of each other given the state, while each state is conditionally independent of all other states given knowledge of the previous state's instantiation.

This simple Markov structure makes it possible to devise learning and inference algorithms to solve problems of research interest,

be it the efficient computation of the probability of the observation sequence $x_{1:d}$ (given knowledge of parameters and state-sequence $s_{1:d}$, where d is the total number of observed data-points in the sequence) or the estimation of $s_{1:d}$ that “best-explains” the observations. In the current study, we are primarily interested in the latter problem. For instance, in the case of speech processing, one would like to find the sequence of words (= states) that most likely generated the speech signal (= observation sequence) recorded by a microphone. When one knows or can accurately estimate (a) the number of possible states K , (b) the transition probabilities, and (c) the functional forms and parameters of the observation pdfs for each state, one can use the Viterbi algorithm to solve this problem [3]. This method is a dynamic programming approach to maximizing the probability $P(s_{1:d} | x_{1:d}, \psi, \theta)$, and has been very successful in practice. Variations of this approach exist to incorporate more complicated dynamics such as autoregressive (AR) processes. When (b) and (c) are not known, it is still possible to use the Viterbi algorithm via a cross-validation/tuning approach [3].

Recently, several efforts have been directed towards solving the same problem in the case when the total number of states is not known or may be potentially infinite [6, 7, 5, 8]. For instance, financial time series data may be governed by any one of a variety of statistical regimes, depending on the vagaries of the wider world. Additionally, one needs to incorporate the fact that new regimes may arise at any time. Nonparametric Bayesian (NPB) methods offer an elegant solution to such problems by using hierarchical Dirichlet process (DP) priors for state transition probability matrices. By construction, these are defined over countably infinite supports, while the hierarchical structure still allows for a finite probability of transition between any pair of states. This allows users to devise efficient Markov chain Monte Carlo (MCMC) sampling schemes to get good representations of the posterior probabilities of the state-sequences, the most common approaches being variations of Gibbs sampling [9] or Metropolis-Hastings methods [10]. These approaches work as long as the Markov structure of the state-evolution is maintained, which means that more complex observation models like linear dynamical systems (LDS) can be utilized without much overhead [5].

In the specific case of switching vector autoregressive (VAR) processes with unknown number of states, NPB methods [5, 11] have been formulated to sample the state-sequences, parameters and transition probabilities of the HMM. One highly effective segmentation method reported in [5] is a variant of the forward-backward (FB) algorithm that uses a truncated approximation of the HDP as a prior for the transition matrix parameters and then samples, at each iteration, the entire state sequence $s_{1:d}$ from $P(s_{1:d} | \psi, \theta, x_{1:d})$. In contrast to the simpler direct Gibbs (DG) approach, which samples each s_j from its full conditional $P(s_j | s_{\setminus j}, x_{1:d}, \theta, \psi)$, this block-sampling method completely uncouples the state-sequence samples in iterations t and $t - 1$ [12]. This leads to faster mixing of the

P. M. Djurić was supported by NSF under Award CCF-1320626.

Markov chain, and gives better representations of the posterior after the burn-in period. However, the segmentation error between the state-sequence sample and the true sequence still has considerable spread. In the present study, we have used a variant of the FB algorithm, where we approximately integrate out some or all of the variables in the set $\{\psi, \theta\}$; in other words, we try to Rao-Blackwellize this sampling strategy. The approximation is necessary because it is not possible to run an FB procedure in finding the sampling probability $P(s_{1:d}|x_{1:d})$ after integrating out the parameters analytically. Instead, we implement integration after the FB procedure is executed many times with different parameter values drawn from their posterior pdf.

In the sequel, we describe the problem formulation, priors and sampling methods in detail in Section 2. We then provide, in Section 3, results of applying this method to two simulated time series that have switching AR dynamics, followed by a discussion of the advantages and potential pitfalls of this method in Section 4.

2. METHODS

We focus, for now, on switching AR models for scalar continuous-time observations, and assume that the state-evolution is governed by a Markov chain, i.e.,

$$P(s_j = k|s_{j-1} = l) = \psi_{l,k}, \quad x_j = \sum_{p=1}^{\rho} A_{k,p} x_{j-p} + w_j,$$

where $A_{k,p}$ denotes, for state k , the AR coefficients, ρ the known model order and w the driving noise, which we assume has a Gaussian pdf with zero mean and known variance σ^2 . Just as in [5], we assume that the A_k s are distributed according to multivariate Gaussian pdfs with hyperparameters μ_{A_k}, Σ_{A_k} . In order to allow for possibly infinite number of states, we can use the hierarchical Dirichlet process (HDP) as a prior for the rows of the transition matrix ψ , i.e.,

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{A_k}, \quad \beta|\gamma \sim \text{GEM}(\gamma),$$

$$G_k = \sum_{l=1}^{\infty} \psi_{k,l} \delta_{A_l}, \quad \psi_k|\alpha, \beta, \kappa \sim \text{DP}\left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_k}{\alpha + \kappa}\right),$$

$$A_k|\mu_A, \Sigma_A \sim \mathcal{N}(\mu_A, \Sigma_A), \quad \forall k \in \{1, 2, \dots\},$$

where β_k denotes the global probability mass associated with the state k and is distributed as a Griffiths-Engen-McCloskey (GEM) process with hyperparameter γ (also called a stick-breaking process) [6]. Thus, G_0 represents the prior pmf of the global frequency of each state. Similarly, G_k represents the prior probability of transition from state k to all other states, and is itself a Dirichlet process with concentration parameter α and base distribution β . This hierarchical construction ensures that each possible state-transition has a finite probability because the global DP G_0 essentially “ties together” each state-associated DP G_k . Additionally, each self transition $k \rightarrow k$ is assumed to have an extra prior probability mass κ , which is called the stickiness mass and encourages the samplers to learn models that have persistence (since real-world data usually exhibit slower transition dynamics). The variables α, γ and κ are also provided their own Gamma priors, and are learned from the data as in [11]. Henceforth, we will refer to the set of hyperparameters $\{\beta, \gamma, \alpha, \mu_A, \Sigma_A\}$ as Φ .

2.1. Direct Gibbs (DG) sampling

The goal of Gibbs sampling in the context of time-series segmentation is to get a representation of the required posterior distribution of

the state-sequence, i.e., to estimate $P(s_{1:d}|x_{1:d}, \Phi)$. For each iteration t , a typical sampling scheme would be, (symbols with superscript (t) denote the value of the corresponding variable at iteration t):

1. Sample transition and AR parameters ψ and A from their conditionals $P(\psi^{(t)}, A^{(t)}|x_{1:d}, s_{1:d}^{(t-1)}, \Phi^{(t-1)})$.
2. Sample the state sequence $s_{1:d}$ from $P(s_{1:d}^{(t)}|x_{1:d}, \psi^{(t)}, A^{(t)})$ (either in a block or sequential fashion).
3. Sample the hyperparameters Φ from the conditional pdf $P(\Phi^{(t)}|x_{1:d}, s_{1:d}^{(t)}, A^{(t)}, \psi^{(t)})$.
4. Update the sufficient statistics and obtain updated posteriors for ψ and A .

In the direct Gibbs sampler, in step 2, we sample each state from its full conditional distribution, which factors as

$$P(s_j^{(t)} = l \mid s_{j-1}^{(t)} = k, s_{j+1}^{(t-1)} = m, x_{1:d}, \psi^{(t)}, A^{(t)}) \propto \psi_{k,l}^{(t)} \psi_{l,m}^{(t)} L(x_j|x_{j-1:j-\rho}, A_l^{(t)}, \sigma^2).$$

This approach necessarily couples the samples at adjacent iterations, which, compounded by the presence of correlated observations from AR processes, results in very slowly mixing Markov chains. Thus, an alternative method, which block-samples the entire state sequence from the pdf $P(s_{1:d}^{(t)}|x_{1:d}, A)$, is used here (as in [5]). We call it the full FB algorithm to distinguish it from our Monte Carlo Rao-Blackwellized version described later.

2.2. Full FB algorithm

The FB sampler is a modification of the forward-backward method of [3] that was originally used for inferring the most probable state-sequences in an HMM. Since the original FB recursions work only for finite state supports, we have to limit the maximum number of possible states to $K'(> K)$, where K is the true number of unique states. The resulting prior is a hierarchical sticky Dirichlet distribution, and is a finite approximation to the sticky-HDP:

$$P(\beta|\gamma) \sim \text{Dirichlet}(\beta; \gamma/K', \dots, \gamma/K'),$$

$$P(\psi_k|\beta) \sim \text{Dirichlet}(\psi_k; \alpha\beta_1, \dots, \alpha\beta_k + \kappa, \dots, \alpha\beta_{K'}).$$

Given $\psi^{(t)}, A^{(t)}$, the FB method for sampling $s_{1:d}$ proceeds as follows:

1. We initialize an array of messages $m_{j,j-1}(k)$ to 1.
 2. We compute, $\forall j \in d, d-1, \dots, 1$ and $\forall k \in 1, \dots, K'$,
- $$m_{j,j-1}(k) := \sum_{l=1}^{K'} \psi_{k,l}^{(t)} m_{j+1,j}(l) \mathcal{N}(x_j; A_k^{(t)} x_{j-1:j-\rho}, \sigma^2).$$
3. We now initialize state-transition counts $z_{k,l} = 0, \forall k, l \in \{1, \dots, K'\}$. For each k , we compute the probability $L_k(x_j) = \mathcal{N}(x_j; A_k^{(t)} x_{j-1:j-\rho}, \sigma^2) m_{j+1,j}(k)$.
 4. We then sample a state assignment

$$s_j^{(t)} \sim \sum_{k=1}^{K'} \psi_{s_{j-1},k}^{(t)} L_k(x_j) \delta(s_j, k),$$

and increment transition counts $z_{k,l}$ accordingly.

The state transition counts are used as sufficient statistics (in addition to the observation sufficient statistics) to update the posteriors of ψ, A, Φ . Details of these computations have been provided in [11].

2.3. FB with Monte-Carlo-based Rao-Blackwellization (RBFB)

The Rao-Blackwell theorem [13, 14] suggests that, if one were to integrate out the “nuisance” parameters ψ, A from the joint distribution, one would obtain samplers that would give more accurate representations of the posterior. Analytically marginalizing out some variables from a joint pdf always reduces the spread of any estimate dependent on it. Thus, we would ideally want to run a sampler that uses the conditional $P(s_{1:d}|x_{1:d})$ instead of $P(s_{1:d}|x_{1:d}, \psi, A)$.

Note, however, that in any HMM with Dirichlet priors, analytically marginalizing out ψ, A from the joint pdf leaves us with a Pólya urn process [15], in which the probability of any transition $k \rightarrow l$ depends on the number of times this transition has already occurred. In effect, not sampling (collapsing out) ψ, A makes every pair of states (s_i, s_j) dependent, and we lose the Markov independence structure that enables FB to work when the parameters are instantiated.

In order to overcome this problem, we use a simple Monte-Carlo integration procedure to “approximately” Rao-Blackwellize the FB method. Prior to step 1 (in the full FB procedure of Section 2.2), instead of sampling only once from the conditional probability distribution of the parameters, we sample M times. For each of the M parameter samples, we perform steps 1 to 3 of the FB process separately and store, for each k, m , the obtained $L_k^{(m)}(x_j)$ terms. Finally, in step 4, we sample each state from

$$\begin{aligned} s_j^{(t)} &\sim \sum_{k=1}^{K'} \psi_{s_{j-1}, k}^{(t)} L'_k(x_j) \delta(s_j, k) \\ &= \frac{1}{M} \sum_{k=1}^{K'} \sum_{m=1}^M \psi_{s_{j-1}, k}^{(t, m)} L_k^{(m)}(x_j) \delta(s_j, k). \end{aligned}$$

In this way, we can preserve the advantages offered by the FB approach (faster mixing) while disposing of the layer of stochastic variability introduced by sampling the model parameters.

3. RESULTS

In order to test the RBFB method, we constructed two simulated time series from switching AR processes of order 1 (Dataset 1) and 2 (Dataset 2) respectively. For each dataset, the true number of states was $K = 3$. Exactly one time series, with $d = 1000$, was generated from each model.

In **Dataset 1**, we fixed the noise variance σ^2 at 0.04 and the true AR parameters to $A_1 = -0.9, A_2 = 0.2, A_3 = 0.9$. The diagonal elements of the true transition probability matrix had much higher values than the non-diagonal ones in order to simulate persistent state dynamics. This matrix was fixed at

$$\psi = \begin{pmatrix} 0.9900 & 0.0051 & 0.0049 \\ 0.0060 & 0.9896 & 0.0044 \\ 0.0056 & 0.0128 & 0.9817 \end{pmatrix}.$$

In **Dataset 2**, we kept $\sigma^2 = 1$, and the true AR parameters for the three states were $A_1 = [0.49, 0.49], A_2 = [1, -0.5]$, and $A_3 = [-1, -0.5]$. The transition probability matrix was fixed at

$$\psi = \begin{pmatrix} 0.9770 & 0.0037 & 0.0193 \\ 0.0085 & 0.9889 & 0.0026 \\ 0.0029 & 0.0564 & 0.9408 \end{pmatrix}.$$

Each time series x was then used as input to the three different Gibbs samplers. Each sampler was run for $T = 3000$ iterations, with each iteration containing 100 hyperparameter sampling steps. When using the RBFB method, we sampled the $\{\psi, A\}$ parameters a total

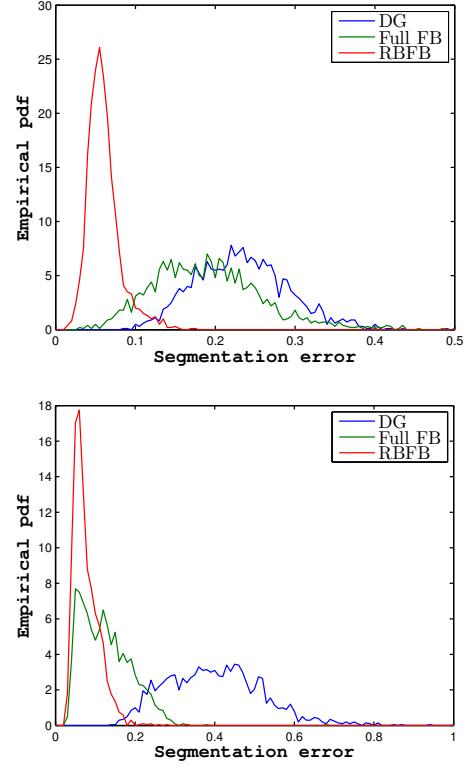


Fig. 1: Empirical pdfs of the segmentation error between sampled and true state-sequences, when using various sampling strategies on the (top) 1st and (bottom) 2-nd order AR data. SE was calculated as explained in the text.

of $M = 10$ times at the start of every iteration. The $\alpha + \kappa$ and γ hyperparameters were both given the same Gamma priors, with the A and B parameters for both priors fixed at 1 and 0.01, respectively. Additionally, the hyperparameter $\kappa/(\alpha + \kappa)$ was used to sample the stickiness parameter, and was given a Beta prior with C and D parameter values of 10 and 1, respectively. Details on updates of the hyperparameters can be found in [11]. For ease of visualization, the maximum number of possible states K' was limited to 5. The state-sequence initialization was $s_j^{(0)} = 1, \forall j = 1, \dots, d (= 1000)$. For the A parameter, a Gaussian prior was used with mean and standard deviation parameters $\mathbf{0}$ and \mathbf{I} , respectively. When sampling A , truncation was enforced in order to ensure that only stable AR processes were sampled [16, 17]. For analysis, after obtaining all the samples, we rejected the first 1000 as burn-in samples.

We calculated the segmentation error (SE) for each iteration’s sampled state sequence in the following way. Since each state is characterized by the corresponding AR parameters, we mapped any given sequence of state-indices $s_{i:j}$ to a sequence of corresponding AR coefficient (or AR coefficient-mean) vectors. For the 1st order AR process, SE for some iteration t is the mean squared error between sequences of AR means obtained from $s_{1:d}^t$ and from the true sequence $s_{1:d}$. For the 2nd-order case, we mapped each state’s AR parameter vector to a set of characteristic-equation roots on the z -plane and calculated the corresponding complex angle. Thus we obtained a sequence of z -plane complex angles for each state-sequence. For instance, a section of the true state-sequence $s_{80:82} = [1, 3, 2]$ was mapped to the A sequence $[[0.49, 0.49], [-1, -0.5], [1, -0.5]]$, then to the z -plane as $[[0.987, -0.497], [-0.5 \pm 0.5i], [0.5 \pm 0.5i]]$, which yielded a se-

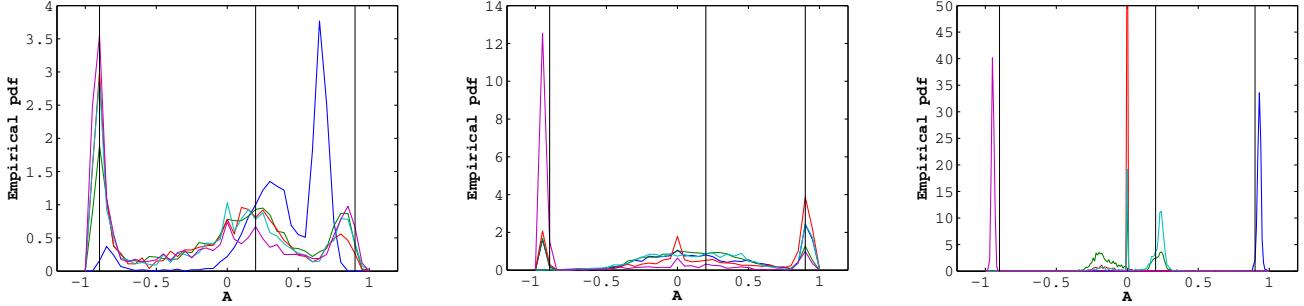


Fig. 2: Empirical pdfs of posterior updates at the end of each iteration for mean of AR parameters for each of the $K' = 5$ possible states ($k = 1, 2, 3, 4, 5$) considered in (left) the DG, (center) the full FB and (right) RBFB sampling strategies. Data used were from dataset 1. Solid black lines indicate the true AR coefficient values.

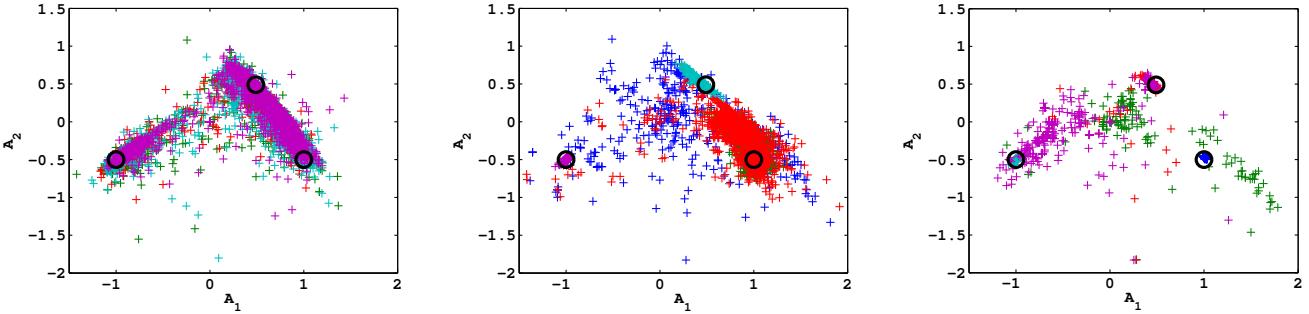


Fig. 3: Posterior updates at the end of each iteration for mean of AR parameters for each of the $K' = 5$ possible states ($k = 1, 2, 3, 4, 5$) considered in (left) the DG, (center) the full FB and (right) RBFB sampling strategies. Data used were from a 2nd-order AR process. Black circles represent the true AR coefficients for the three states. Note that, in the case of RBFB, the cyan and blue markers are isolated in very small regions close to the true parameter values (1,0.5) and (-1,-0.5), respectively.

quence of complex angles (in radians) $[0, 2.356, 0.785]$. Similarly, we obtained sequences of z -plane angles for each iteration's sampled state-sequence $s_{1:d}^{(t)}$ and calculated the mean squared error between this and the true z -plane argument sequence. Figure 1 shows comparisons of the empirical pdfs constructed from the histograms of the SEs obtained from the DG, full FB and RBFB strategies' respective samples after burn-in. Using the full FB method yields lower SE on average compared to DG, and the RBFB further improves on this.

Figure 2 shows empirical pdfs constructed from histograms of parameter mean updates at the end of each sampling iteration (i.e., after step 4 in Section 2.1) for each state considered in the DG, full FB and RBFB strategies when applied to Dataset 1. Similarly, for Dataset 2, we plot in Fig. 3 the parameter means obtained at the end of each iteration from the three samplers. Note that in the right panel of Fig. 2, which shows results for the RBFB method, the means for state 3 (in red) are concentrated around 0. This is because for almost all iterations, this state was not instantiated when sampling for $s_{1:d}$, and the sampler generates A values from the prior. The same is true for state 2 (cyan) for a smaller number of iterations. In both figures, the parameter means are scattered around the true AR coefficients, and the RBFB sampler yields the least spread of all the samplers, while accurately identifying the three existing AR models in most iterations.

4. DISCUSSION

NPB sampling methods have been steadily gaining in popularity to solve inference problems, aided by improved understanding of the theory and the potency of present day computers for their implementation [18, 19]. Interesting applications include text classification [6], image processing [4], adult heart rate data analysis [20, 21], and many more. In these methods, the number of classes need not be predefined. During sampling, when new data are processed, they

can either join one of the existing classes or can be grouped in a new class. This allows the number of classes to grow as new data are acquired, while priors such as Dirichlet processes enforce model compactness implicitly.

Considerable attention has also been devoted to NPB segmentation of switching autoregressive processes, notably by [5, 6, 8]. In [5], the authors have shown the use of a Baum-Welch type FB recursion to block-sample the state sequence of the HMM ($s_{1:d}$) from $P(s_{1:d}^{(t)}|x_{1:d}, \psi^{(t)}, A^{(t)})$ in order to increase the mixing rates in the direct Gibbs (DG) approach. In this study, we improved upon these results by Rao-Blackwellizing the FB algorithm using a Monte-Carlo approach. By sampling ψ, A , multiple times from their respective posteriors and averaging the resultant posterior probability of the state-sequence, we are able to encourage the sampler to explore the posterior landscape more efficiently. Our simulations with two different models of AR order 1 and 2, suggest that the RBFB sampler can decrease the mean and spread of the segmentation errors over many iterations (Fig. 1). In Fig. 2, one can see that when using the DG method on 1-st order AR data, each considered state wandered through a wide region in the A space. Also, while the $A = -0.9$ state was identified in most iterations by all samplers, none of the considered states in the DG method could uniquely pick out the $A = 0.9$ or $A = 0.2$ states. When using the full FB method, it is clear that the true states 1 and 3 are picked out by more than one of the considered states, while the $A = 0.2$ state is ignored. Finally, the RBFB sampler yields much narrower empirical pdfs for each considered state, and is the only one that is able to pick out the $A = 0.2$ state consistently. Moreover, for most iterations, each state considered by the RBFB samples from unique regions in the A space, leading to less uncertainty in state-identification. Similar results can be seen for Dataset 2 in Fig. 3. In particular, for the RBFB method, we note that the narrow region in the A_1, A_2 space

occupied by states 1 and 4 is very close to the true A vectors (1,-0.5) and (-1,-0.5) (black circles).

One limitation of the RBFB method is the significant overhead in terms of computational complexity; the present method performs the entire sampling chain M times for each sampled value of the parameter set, which increases the time taken for sampling M times. However, vectorizing the code (programs were implemented in Matlab) for the parameter sampling step enabled us to make the RBFB method more efficient. We performed some small simulations to test the computational savings from vectorization. Using observed time-series from Dataset 1, we found that on average, the vectorized RBFB method completed the entire sampling chain in about 1.5 s per iteration, whereas the code without vectorization took about 2.5 s per iteration on average. As expected, the direct Gibbs and FB methods outperformed the RBFB method in computation time. However, we note that further studies need to be done to analyze the computational complexity of the samplers, and to decrease it if possible. In addition, we are in the process of testing the effect of Rao-Blackwellization on the segmentation error with different model complexities (higher order AR or more general linear dynamical systems), higher noise variance and for varying numbers of M .

5. REFERENCES

- [1] P. M. Djurić, S. M. Kay, and G. F. Boudreault-Bartels, “Segmentation of nonstationary signals,” in *Acoustics, Speech, and Signal Processing, 1992 IEEE International Conference on*, vol. 5, 1992, pp. 161–164.
- [2] R. J. Boys and D. A. Henderson, “A Bayesian approach to DNA sequence segmentation,” *Biometrics*, vol. 60, no. 3, pp. 573–581, 2004.
- [3] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [4] E. Sudderth, “Graphical models for visual object recognition and tracking,” Ph.D. dissertation, Massachusetts Institute of Technology, 2006.
- [5] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, “Bayesian nonparametric inference of switching dynamic linear models,” *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1569–1585, 2011.
- [6] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Hierarchical Dirichlet processes,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.
- [7] J. K. Ghosh and R. V. Ramamoorthi, *Bayesian Nonparametrics*. Springer, 2003.
- [8] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, “The infinite hidden Markov model,” in *Advances in Neural Information Processing Systems*, 2001, pp. 577–584.
- [9] P. Resnik and E. Hardisty, “Gibbs sampling for the uninitiated,” DTIC Document, Tech. Rep., 2010.
- [10] R. M. Neal, “Markov chain sampling methods for Dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.
- [11] E. B. Fox, “Bayesian nonparametric learning of complex dynamical phenomena,” Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
- [12] S. L. Scott, “Bayesian methods for hidden Markov models: Recursive computing in the 21st century,” *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 337–351, 2002. [Online]. Available: <http://www.jstor.org/stable/3085787>
- [13] S. M. Kay, *Fundamentals of Statistical Signal Processing*, A. V. Oppenheim, Ed. Prentice-Hall, 1993.
- [14] J. S. Liu, W. H. Wong, and A. Kong, “Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes,” *Biometrika*, vol. 81, no. 1, pp. 27–40, 1994.
- [15] D. Blackwell and J. B. MacQueen, “Ferguson distributions via Pólya urn schemes,” *The Annals of Statistics*, pp. 353–355, 1973.
- [16] P. Shcherbakov and F. Dabbene, “On random generation of stable polynomials,” in *Control Applications,(CCA) & Intelligent Control,(ISIC), 2009 IEEE*. IEEE, 2009, pp. 406–411.
- [17] E. R. Beadle and P. M. Djurić, “Uniform random parameter generation of stable minimum-phase real ARMA(p, q) processes,” *Signal Processing Letters, IEEE*, vol. 4, no. 9, pp. 259–261, 1997.
- [18] P. Müller and R. Mitra, “Bayesian nonparametric inference: Why and How,” *Bayesian Analysis*, vol. 8, no. 2, pp. 1–34, 2013.
- [19] N. L. Hjort, *Bayesian Nonparametrics*. Cambridge University Press, 2010, no. 28.
- [20] A. Van Esbroeck, C.-C. Chia, and Z. Syed, “Heart rate topic models,” in *The Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 1001, 2012, p. 48109.
- [21] S. Saria, D. Koller, and A. Penn, “Learning individual and population level traits from clinical temporal data,” in *Proc. Neural Information Processing Systems (NIPS), Predictive Models in Personalized Medicine workshop*, 2010.