Journal of Machine Learning Research 18 (2017) 1-29 Submitted 11/15; Revised 7/17; Published 8/17

Variational Particle Approximations

Ardavan Saeedi* ARDAVANS@MIT.EDU
Computer Science & Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Tejas D. Kulkarni* TEJASDKULKARNI@GMAIL.COM
DeepMind, London

Vikash K. Mansinghka VKM@MIT.EDU
Department of Brain & Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Samuel J. Gershman GERSHMAN@FAS.HARVARD.EDU
Department of Psychology and Center for Brain Science
Harvard University
Cambridge, MA 02138, USA

Editor: Lawrence Carin

Abstract

Approximate inference in high-dimensional, discrete probabilistic models is a central problem in computational statistics and machine learning. This paper describes discrete particle variational inference (DPVI), a new approach that combines key strengths of Monte Carlo, variational and search-based techniques. DPVI is based on a novel family of particle-based variational approximations that can be fit using simple, fast, deterministic search techniques. Like Monte Carlo, DPVI can handle multiple modes, and yields exact results in a well-defined limit. Like unstructured mean-field, DPVI is based on optimizing a lower bound on the partition function; when this quantity is not of intrinsic interest, it facilitates convergence assessment and debugging. Like both Monte Carlo and combinatorial search, DPVI can take advantage of factorization, sequential structure, and custom search operators. This paper defines DPVI particle-based approximation family and partition function lower bounds, along with the sequential DPVI and local DPVI algorithm templates for optimizing them. DPVI is illustrated and evaluated via experiments on lattice Markov Random Fields, nonparametric Bayesian mixtures and block-models, and parametric as well as non-parametric hidden Markov models. Results include applications to real-world spike-sorting and relational modeling problems, and show that DPVI can offer appealing time/accuracy trade-offs as compared to multiple alternatives.

Keywords: Bayesian inference, variational methods, Dirichlet process mixture model, Ising model, hidden Markov model, infinite relational model

1. Introduction

Probabilistic models defined over large collections of discrete random variables have arisen in multiple fields. Examples include hidden Markov models for sequential data; Bayesian

* Denotes equal contribution.

networks; mixture models for tabular and relational data; and discrete Markov random field models, which have become popular in fields ranging from computer vision to information extraction. These models are typically specified in terms of a probability distribution $P(x)$ defined over a collection of variables $x = \{x_n\}$ occupying points in an underlying discrete space $\mathcal{X}$. One key approximate inference problem that arises in many applications is identifying high-probability configurations of the discrete variables.

Most approximate inference algorithms fall into two classes: Monte Carlo methods and variational methods. Monte Carlo methods generate samples from approximations to the posterior distribution that grow more accurate as the technique is given more compute time. The flexibility and simplicity of Monte Carlo methods have made them the workhorse of statistical computation (Robert and Casella, 2004). There are two basic approaches to Monte Carlo inference for discrete probabilistic models. The first, Markov chain Monte Carlo methods, work by running a Markov chain with transition operator T whose equilibrium distribution asymptotically approaches $P(x)$—that is, $T^k P_0(x) \approx P(x)$ for any initial distribution on states $P_0(x)$. The second, sequential Monte Carlo methods, build up a sample from a distribution that approximates $P(x)$ by sampling from a sequence of more tractable distributions, typically defined over subspaces of $\mathcal{X}$. However, their accuracy is difficult to measure, and the amount of computation required for satisfactory accuracy can be prohibitive in practice.

Variational methods (Wainwright and Jordan, 2008) treat probabilistic inference as an optimization problem over a set of distributions. This set is typically constrained (e.g., to factorized conjugate exponential distributions), thereby attaining efficiency at the expense of bias. In particular, variational methods tend to converge quickly and supply an easily monitored objective function (unlike Monte Carlo methods). However, for complex discrete models, the bias induced by variational approximations can sometimes lead to poor predictive performance. For example, consider a discrete probabilistic model where two binary variables x_1 and x_2 are constrained to take the same value, i.e. $P(x_1, x_2) = 0$ if $x_1 \neq x_2$. All the probability mass is on the states $x_1 = x_2 = 0$ and $x_1 = x_2 = 1$. In "mean-field" variational inference, this distribution might be approximated as $Q_{\theta_1}(x_1)Q_{\theta_2}(x_2)$, with θ_1 and θ_2 representing coin weights that used to model x_1 and x_2 as independent Bernoulli distributions. This family of variational approximations to $P(x_1, x_2)$ cannot capture the true distribution, and in fact cannot qualitatively capture the simple constraint that $x_1 = x_2$. These limitations prompted the development of more sophisticated approximations (e.g., Bouchard-Côté and Jordan, 2009; Jaakkola and Jordan, 1998), but these incur additional computational cost and can be difficult or impossible to apply to a given problem.

This paper introduces a new approximate inference method, called *discrete particle variational inference* (DPVI), that aims to combine key strengths of both Monte Carlo and variational inference. The key insight in DPVI is to use a weighted collection of samples—the kind of "particle approximation" output by Monte Carlo methods—as the approximating family for variational inference. Suppose we got to pick where to place the particles in the hypothesis space; where would we put them? Intuitively, we would want to distribute them in such a way that they cover high probability regions of the target distribution, but without the particles all devolving onto the mode of the distribution. This problem can be formulated precisely within the framework of variational inference. We derive a coordinate

ascent update for particle approximations that iteratively minimizes the Kullback-Leibler (KL) divergence between the particle approximation and the target distribution.

In DPVI, the location of the particles become the parameters of the approximating family. This simple choice has appealing consequences. Like Monte Carlo, DPVI can handle problems where the posterior has multiple modes, and yields exact results in a well-defined limit (as the number of particles goes to infinity). Like standard mean-field variational methods, DPVI is based on optimizing a lower bound on the partition function; when this quantity is not of intrinsic interest, it facilitates convergence assessment and debugging. Like both Monte Carlo and combinatorial search, DPVI can take advantage of factorization, sequential structure, and custom search operators.

The rest of the paper is organized as follows. After introducing our general framework, we describe how it can be applied to filtering and smoothing problems. We then show experimentally that variational particle approximations can overcome a number of problems that are challenging for conventional Monte Carlo methods. In particular, our approach is able to produce a diverse, high probability set of particles in situations where Monte Carlo and mean-field variational methods sometimes degenerate.

2. Background

Consider the problem of approximating a probability distribution $P(x)$ over discrete latent variables $x = \{x_1, \ldots, x_N\}, x_n \in \{1, \ldots, M_n\}$, where the target distribution is known only up to a normalizing constant Z: $P(x) = f(x)/Z$. We will refer to $f(x) \geq 0$ as the *score* of x and Z as the *partition function*. We further assume that $P(x)$ is a Markov network defined on a graph G, so that $f(x)$ factorizes according to:

$$f(x) = \prod_c f_c(x_c), \tag{1}$$

where $c \subseteq \{1, \ldots, N\}$ indexes the maximal cliques of G.

2.1 Importance sampling and sequential Monte Carlo

A general way to approximate $P(x)$ is with a weighted collection of K particles, $\{x^1, \ldots, x^K\}$:

$$P(x) \approx Q(x) = \sum_{k=1}^{K} w^k \delta[x, x^k], \tag{2}$$

where $x^k = \{x_1^k, \ldots, x_N^k\}, x_n^k \in \{1, \ldots, M_n\}$ and $\delta[\cdot, \cdot] = 1$ if its arguments are equal and 0 otherwise. Importance sampling is a Monte Carlo method that stochastically generates particles from a proposal distribution, $x^k \sim \phi(\cdot)$, and computes the weight according to $w^k \propto f(x^k)/\phi(x^k)$. Importance sampling has the property that the particle approximation converges to the target distribution as $K \to \infty$ (Robert and Casella, 2004).

Sequential Monte Carlo methods such as particle filtering (Doucet et al., 2001) apply importance sampling to stochastic dynamical systems (where n indexes time) by sequentially sampling the latent variables at each time point using a proposal distribution $\phi(x_n|x_{n-1})$. This procedure can produce conditionally low probability particles; therefore, most algorithms include a resampling step which replicates high probability particles and kills off

low probability particles. The downside of resampling is that it can produce degeneracy: the particles become concentrated on a small number of hypotheses, and consequently the effective number of particles is low.

2.2 Variational Inference

Variational methods (Wainwright and Jordan, 2008) define a parametrized family of probability distributions $\mathcal{Q}$ and then choose $Q \in \mathcal{Q}$ that maximizes the *negative variational free energy*:

$$\mathcal{L}[Q] = \sum_x Q(x) \log \frac{f(x)}{Q(x)}. \tag{3}$$

The negative variational free energy is related to the partition function Z and the KL divergence through the following identity:

$$\log Z = \mathrm{KL}[Q||P] + \mathcal{L}[Q], \tag{4}$$

where

$$\mathrm{KL}[Q||P] = \sum_x Q(x) \log \frac{Q(x)}{P(x)}. \tag{5}$$

Since $\mathrm{KL}[Q||P] \geq 0$, the negative variational free energy is a lower bound on the log partition function, achieving equality when the KL divergence is minimized to 0. Maximizing $\mathcal{L}[Q]$ with respect to Q is thus equivalent to minimizing the KL divergence between Q and P.

Unlike the Monte Carlo methods described in the previous section, variational methods do not in general converge to the target distribution, since typically P is not in $\mathcal{Q}$. The advantage of variational methods is that they guarantee an improved bound after each iteration, and convergence is easy to monitor (unlike most Monte Carlo methods). In practice, variational methods are also often more computationally efficient.

We next consider particle approximations from the perspective of variational inference. We then turn to the application of particle approximations to inference in stochastic dynamical systems.

3. Variational Particle Approximations

Variational inference can be connected to Monte Carlo methods by viewing the particles as a set of variational parameters parameterizing Q. For the particle approximation defined in Eq. 2, the negative variational free energy takes the following form:

$$\mathcal{L}[Q] = \sum_{k=1}^{K} w^k \log \frac{f(x^k)}{w^k V^k}, \tag{6}$$

where $V^k = \sum_{j=1}^{K} \delta[x^j, x^k]$ is the number of times an identical replica of x^k appears in the particle set. We wish to find the set of K particles and their associated weights that maximize $\mathcal{L}[Q]$, subject to the constraint that $\sum_{k=1}^{K} w^k = 1$. This constraint can be implemented

by defining a new functional with Lagrange multiplier λ:

$$\tilde{\mathcal{L}}[Q] = \mathcal{L}[Q] + \lambda \left(\sum_{k=1}^{K} w^k - 1 \right). \tag{7}$$

Taking the functional derivative of the Lagrangian with respect to w^k and equating to zero, we obtain:

$$\begin{aligned} \frac{\partial \tilde{\mathcal{L}}[Q]}{\partial w^k} &= \log f(x^k) - \log w^k - \log V^k + \lambda - 1 = 0 \\ &\implies w^k = Z_Q^{-1} f(x^k)/V^k, \end{aligned} \tag{8}$$

where

$$Z_Q = \exp(\lambda - 1)^{-1} = \sum_{k=1}^{K} \frac{f(x^k)}{V^k}. \tag{9}$$

We can plug the above result back into the definition of $\mathcal{L}[Q]$:

$$\begin{aligned} \mathcal{L}[Q] &= Z_Q^{-1} \sum_{k=1}^{K} \frac{f(x^k)}{V^k} \log \frac{f(x^k)V^k}{Z_Q^{-1} f(x^k) V^k} \\ &= Z_Q^{-1} \sum_{k=1}^{K} \frac{f(x^k)}{V^k} \log Z_Q \\ &= \log Z_Q. \end{aligned} \tag{10}$$

Thus, $\mathcal{L}[Q]$ is maximized by choosing the K values of x with the highest score. The following theorem shows that allowing $V^k > 1$ (i.e., having replica particles) can never improve the bound.

Theorem 1 *Let Q and Q' denote two particle approximations, where Q consists of unique particles ($V_Q^k = 1$ for all k) and Q' is identical to Q except that particle $x_{Q'}^j$ is replicated $V_{Q'}^j$ times (displacing $V_{Q'}^j$ other particles with cumulative score F). For any choice of particles, $\mathcal{L}[Q] \geq \mathcal{L}[Q']$.*

Proof We first apply Jensen's inequality to obtain an upper bound on $\mathcal{L}[Q']$:

$$\mathcal{L}[Q'] \leq \log \sum_{k=1}^{K} w_{Q'}^k Z_{Q'} = \log \sum_{k=1}^{K} \frac{f(x_{Q'}^k)}{V_{Q'}^k}. \tag{11}$$

Since $\mathcal{L}[Q] = \log Z_Q$, we wish to show that $Z_Q \geq \sum_{k=1}^{K} \frac{f(x_{Q'}^k)}{V_{Q'}^k}$. All the particles in Q and Q' are identical except for the $V_{Q'}^j$ particles in Q that were displaced by replicas of $x_{Q'}^j$ in Q'; thus we only need to establish that $f(x_{Q'}^j) + F \geq \frac{V_{Q'}^j f(x_{Q'}^j)}{V_{Q'}^j} = f(x_{Q'}^j)$, where the left hand side of the inequality is the change in negative variational free energy after the replication

of particles. Since the score $f(x)$ can never be negative, the cumulative score F can also never be negative ($F \geq 0$) and the inequality holds for any choice of particles. ■

The variational bound can be optimized by coordinate ascent, as specified in Algorithm 1, which we refer to as *discrete particle variational inference* (DPVI). This algorithm takes advantage of the fact that when optimizing the bound with respect to a single variable, only potentials local to that variable need to be computed. In particular, let $\tilde{x}^k$ be a replica of x^k with a single-variable modification, $\tilde{x}^k_n = m$. We can compute the unnormalized probability of this particle efficiently using the following equation:

$$f(\tilde{x}^k) = f(x^k)\frac{\mathcal{F}_n(\tilde{x}^k)}{\mathcal{F}_n(x^k)}, \tag{12}$$

where $\mathcal{F}_n(x) = \prod_{c:n\in c} f_c(x_c)$. The variational bound for the modified particle can then be computed using Eq. 10. Particles can be initialized arbitrarily. When repeatedly iterated, DPVI will converge to a local maximum of the negative variational free energy.[1] Note that in principle more sophisticated methods can be used to find the top K modes (e.g., Flerova et al., 2012; Yanover and Weiss, 2003); however, we have found that this coordinate ascent algorithm is fast, easy to implement, and very effective in practice (as our experiments below demonstrate).

An important aspect of this framework is that it maintains one of the same asymptotic guarantees as importance sampling: Q converges to P as $K \to \infty$, since in this limit DPVI is equivalent to exact inference. Thus, DPVI combines advantages of variational methods (monotonically decreasing KL divergence between Q and P) with the asymptotic correctness of Monte Carlo methods. It is important to note that asymptotic correctness might be useless in practice unless something is known about the convergence rate. This issue is not unique to DPVI; it also applies to Monte Carlo and variational methods. For certain Markov chain Monte Carlo (MCMC) samplers, it can be shown that the chain converges to the posterior at a geometric rate (Mengersen et al., 1996; Meyn and Tweedie, 1993). A small amount of work has investigated convergence properties of variational methods for specific models (Hall et al., 2011; Wang et al., 2006), but in general the issue of convergence rate for variational methods is an open question.

The asymptotic complexity of DPVI in the sequential setting is $O(SNK)$ where S is the maximum support size of the latent variables. For the iterative update of the particles the complexity is $O(TCSK)$, where T is the maximum number of iterations until convergence and C is the maximum clique size. In our experiments, we empirically observed that we only need a small number of iterations and particles in order to outperform our baselines.

1. Naturally, initialization affects performance, since the objective function has local optima. For example, if the posterior is multimodal and none of the particles are initialized near the dominant mode, then the particle approximation will likely miss a significant portion of the probability mass. Studying the effects of initialization is an important practical challenge for the application of DPVI. In our experiments, we report averages across multiple random initializations.

Algorithm 1 Discrete particle variational inference

/*N is the number of latent variables */
/*x^k is the set of all latent variables for the kth particle: $x^k = \{x_1^k, \ldots, x_N^k\}$ */
/*M_n is the support of latent variable x_n */
Input: initial particle approximation Q with K particles, tolerance ϵ
while $|\mathcal{L}[Q] - \mathcal{L}[Q']| > \epsilon$ **do**
  for $n = 1$ **to** N **do**
    $\mathcal{X} = \emptyset$
    for $k = 1$ **to** K **do**
      Copy particle k: $\tilde{x}^k \leftarrow x^k$
      for $m = 1$ **to** M_n **do**
        Modify particle: $\tilde{x}_n^k \leftarrow m$
        Score $\tilde{x}^k$ using Eq. 12
        $\mathcal{X} \leftarrow \mathcal{X} \cup (\tilde{x}^k, f(\tilde{x}^k))$
      end for
    end for
    Select K unique particles from $\mathcal{X}$ with the largest scores
    Construct new particle approximation $Q'(x) = \sum_{k=1}^K w^k \delta[x, x^k]$
    Compute variational bound $\mathcal{L}[Q']$ using Eq. 10
  end for
end while
return particle approximation Q'

4. Filtering and Smoothing in Hidden Markov Models

We now describe how variational particle approximations can be applied to filtering and smoothing in hidden Markov models (HMMs). Consider a hidden Markov model with observations $y = \{y_1, \ldots, y_N\}$ generated by the following stochastic process:

$$P(y, x, \theta) = P(\theta) \prod_n P(y_n|x_n, \theta) P(x_n|x_{n-1}, \theta), \tag{13}$$

where θ is a set of transition and emission parameters. We are particularly interested in *marginalized* HMMs where the parameters are integrated out: $P(y, x) = \int_\theta P(y, x, \theta) d\theta$. This induces dependencies between observation n and all previous observations, making inference challenging.

Filtering is the problem of computing the posterior over the latent variables at time n given the history $y_{1:n}$. To construct the variational particle approximation of the filtering distribution, we need to compute the product of potentials for variable n:

$$\mathcal{F}_n(x) = P(y_n|x_{1:n}, y_{1:n-1}) P(x_n|x_{1:n-1}). \tag{14}$$

Recall from the previous section that $\mathcal{F}_n(x)$ is the joint probability of the maximal cliques to which x_n belongs. We can then apply the coordinate ascent update described in the previous section. This update is simplified in the filtering context due to the underlying Markov structure. Specifically, Eq. 12 is given by:

$$f(\tilde{x}^k) = f(x^k) P(y_n|x_n^k = m, x_{1:n-1}^k, y_{1:n-1}) P(x_n^k = m|x_{1:n-1}^k). \tag{15}$$

At each time step, the algorithm selects the K continuations (new variable assignments of the current particle set) that maximize the negative variational free energy.

Smoothing is the problem of computing the posterior over the latent variables at time n given data from both the past and the future, $y_{1:N}$. The product of potentials is given by:

$$\mathcal{F}_n(x) = P(y_n|x_{1:n}, y_{-n})P(x_n|x_{-n}), \tag{16}$$

where x_{-n} refers to all the latent variables except x_n (and likewise for y_{-n}). This potential can be plugged into the updates described in the previous section.

To understand DPVI applied to filtering problems, it is helpful to contemplate three possible fates for a particle at time n (illustrated in Figure 1):

- **Selection**: A single continuation of particle k has non-zero weight. This can be seen as a deterministic version of particle filtering, where the sampling operation is replaced with a max operation.
- **Splitting**: Multiple continuations of particle k have non-zero weight. In this case, the particle is split into multiple particles at the next iteration.
- **Deletion**: No continuations of particle k have non-zero weight. In this case, the particle is deleted from the particle set.

Similar to particle filtering with resampling, DPVI deletes and propagates particles based on their probability. However, as we show later, DPVI is able to escape some of the problems associated with resampling.

5. Related Work

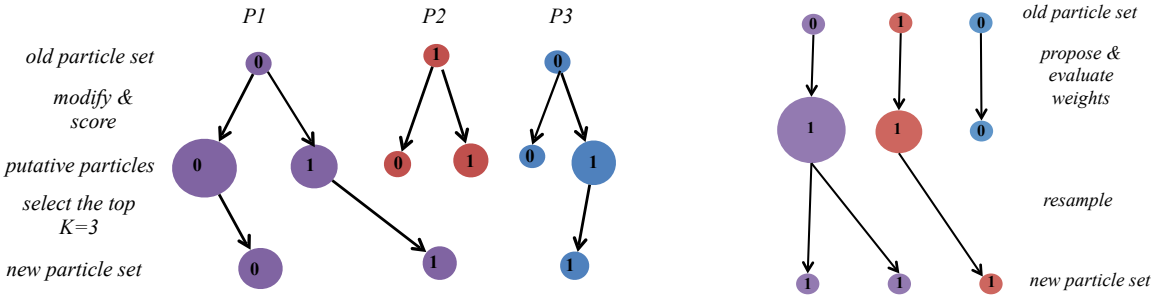
DPVI is related to several other ideas in the statistics literature:

- DPVI is a special case of a *mixture mean-field variational approximation* (Jaakkola and Jordan, 1998; Lawrence, 2000):

$$Q(x) = \sum_{k=1}^{K} Q(k) \prod_{n=1}^{N} Q(x_n|k). \tag{17}$$

  In DPVI, $Q(k) = w^k$ and $Q(x_n|k) = \delta[x_n, x_n^k]$. From the simple restriction that the component distributions must be delta functions, we derived a new algorithm that is simpler and more efficient than mixture mean-field (which requires separate updates for the mixture weights), while sacrificing some of its expressivity. Another distinct advantage of DPVI is that the variational updates do not require the additional lower bound used in general mixture mean-field, due to the intractability of the mean-field updates.

- When $K = 1$, DPVI is equivalent to *iterated conditional modes* (ICM; Besag, 1986), which iteratively maximizes each latent variable conditional on the rest of the variables. This algorithm is simple to implement, efficient (relative to variational and Monte Carlo algorithms), and has been successfully applied to computer vision tasks



(A) DPVI (B) Particle Filtering

Figure 1: Schematic of DPVI versus particle filtering for filtering problems. Illustration of different filtering scenarios over 2 time steps in a binary state space with $K = 3$ particles. The number in each circle indicates the binary value of the corresponding variable. Arrows indicate the evolution of the particles. (A) DPVI: The size of the putative particles represents the score of the particle. The K continuations with highest score are selected for propagation to the next time step. The size of the new particle set corresponds to the normalized score. Particle $P1$ is split, $P2$ is deleted and one putative particle from $P3$ is selected. (B) Particle filtering: The size of the node represents the weight of the particle for the resampling step.

such as reconstruction and segmentation of Markov random fields. However, the algorithm is susceptible to local optima without the aid of relaxation techniques like simulated annealing (Greig et al., 1989).

- DPVI is conceptually similar to nonparametric variational inference (Gershman et al., 2012), which approximates the posterior over a continuous state space using a set of particles convolved with a Gaussian kernel (see Miller et al., 2016, for more sophisticated extensions of this idea). This approach was shown to be effective for probabilistic models that lack the conjugate-exponential structure required for exact mean-field inference. Because nonparametric variational inference approximates continuous densities, it is inapplicable to the discrete problems considered here.

- Frank et al. (2009) used particle approximations within a variational message passing algorithm. The resulting approximation is "local" in the sense that the particles are used to approximate messages passed between nodes in a factor graph, in contrast to the "global" approximation produced by DPVI, which attempts to capture the distribution over the entire set of variables. It is an interesting question for future research to understand what classes of probabilistic models are better approximated using local vs. global approaches.

- Ionides (2008) described a truncated version of importance sampling in which weights falling below some threshold are set to the threshold value. Ionides (2008) showed that truncation reduced sensitivity to the proposal distribution and derived optimal

truncations as a function of the number of samples. This is similar (though not equivalent) to the DPVI setting where latent variables are sampled exhaustively and without replacement.

- Schniter et al. (2008) described an approximate inference algorithm, *fast Bayesian matching pursuit*, which can be viewed as a special case of DPVI applied to Gaussian mixture models.

- In Jones et al. (2005), a *shotgun stochastic search* algorithm is suggested, which proposes local changes to the latent variables with probability proportional to the unnormalized posterior. While this method of evaluating local changes is similar to our coordinate algorithm for the DPVI objective, it is important to note that DPVI is not a stochastic search algorithm (unless $K = 1$): It maintains a collection of particles in order to approximate the posterior distribution, which is important for applications that require a representation of uncertainty.

- Finally, DPVI is closely related to the problem of finding the K most probable latent variable assignments (Flerova et al., 2012; Yanover and Weiss, 2003). We view this problem through the lens of particle approximations, connecting it to both Monte Carlo and variational methods. The techniques developed for finding the K-best assignments could be fruitfully applied to optimizing the DPVI objective function.

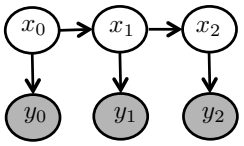
Our experimental strategy is to compare DPVI with popular algorithms that have similar computational complexity, which is why we focus on particle filtering, Gibbs sampling and mean-field approximations. Although mixture mean-field will by definition lead to a better approximation, its computational complexity is considerably greater, which may explain why it is has never achieved the popularity of standard mean-field approaches. Some of the approaches listed above are not applicable to the discrete setting (nonparametric variational inference and fast Bayesian matching pursuit), and some are point estimators instead of true probabilistic approximations (iterated conditional modes and shotgun stochastic search).

6. Experiments

In this section, we compare the performance of DPVI to several widely used approximate inference algorithms, including particle filtering, Gibbs sampling and variational methods. We first present a didactic example to illustrate how DPVI can sometimes succeed where particle filtering fails. We then apply DPVI to four popular but intractable probabilistic models: the Dirichlet process mixture model (DPMM; Antoniak, 1974; Escobar and West, 1995), the infinite HMM (iHMM; Beal et al., 2002; Teh et al., 2006), the infinite relational model (IRM; Kemp et al., 2006) and the Ising model.

(A)

DPVI	Monte Carlo	Mean-Field
$Q(x) = \sum_{k=1}^{K} w^k \prod_{n=1}^{N} \delta[x_n, x_n^k]$	$Q(x) = \frac{1}{K} \sum_{k=1}^{K} \prod_{n=1}^{N} \delta[\hat{x}_n, \hat{x}_n^k]$	$Q(x) = \prod_{n=1}^{N} Q(x_n)$



(B)

DPVI		Monte Carlo	Mean-Field
State Seq. $[x_0, x_1, x_2]$	Weight	State Seq. $[x_0, x_1, x_2]$	Mean-Field Parameters
[0,0,0]	0.15	[1,0,1]	$Q(x_0 = 0) = 0.4$
[0,1,1]	0.03	[1,1,1]	$Q(x_1 = 0) = 0.63$
[1,1,1]	0.32	[1,1,1]	$Q(x_2 = 0) = 0.15$
[1,0,0]	0.15	[1,1,1]	
[1,0,1]	0.35	[1,0,1]	
$Pr(x_0 = 1, x_1 = 0, x_2 = 1\|y) \approx 0.35$		$Pr(x_0 = 1, x_1 = 0, x_2 = 1\|y) \approx 0.4$	$Pr(x_0 = 1, x_1 = 0, x_2 = 1\|y) \approx 0.32$

Figure 2: Comparison of approximate inference schemes. (A) Approximating families for DPVI, Monte Carlo and mean-field. (B) Approximating the posterior probability of a sequence $(x_0 = 1, x_1 = 0, x_2 = 1)$ for the above 3 schemes on a binary HMM. Given the weights for different sequences in DPVI the posterior probability is the weight corresponding to that sequence. For Monte Carlo approximation, the posterior can be approximated from the normalized counts of sampled $(x_0 = 1, x_1 = 0, x_2 = 1)$ sequences. Finally, for the mean-field approximation, we have $Pr(x_0 = 1, x_1 = 0, x_2 = 1|y) = Q(x_0 = 1)Q(x_1 = 0)Q(x_2 = 1)$.

6.1 Didactic Example: Binary HMM

As a didactic example, we use a simple HMM with binary hidden states (x) and observations (y):

$$\begin{aligned} P(x_{n+1} = 0|x_n = 0) &= \alpha_0 \\ P(x_{n+1} = 1|x_n = 1) &= \alpha_1 \\ P(y_n = 0|x_n = 0) &= \beta_0 \\ P(y_n = 1|x_n = 1) &= \beta_1, \end{aligned} \tag{18}$$

with α_0, α_1, β_0, and β_1 all less than 0.5. Constraining the parameters to be less than 0.5 makes some sequences more likely; approximating the posterior using a particle filter (with resampling) may result in capturing only these sequences. We will use this model to illustrate how DPVI differs from particle filtering. Figure 2 compares several inference schemes for this model.

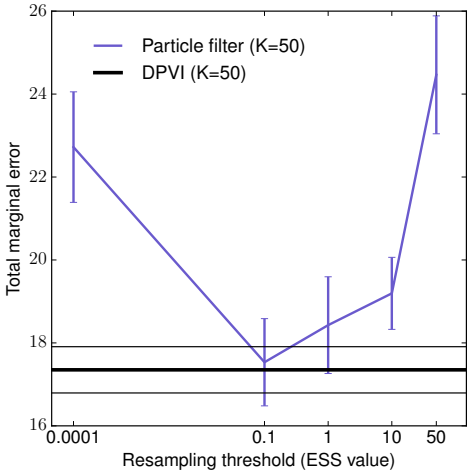


Figure 3: HMM with binary hidden states and observations. Total marginal error computed for a sequence of length 200. For particle filtering the total error for every ESS value is averaged over 5 sequences generated from the HMM; in addition, for each sequence we reran the particle filter 5 times (thus 25 runs total). Note the logarithmic scale of the x-axis. Error bars and the thin black lines correspond to standard error of the mean.

For illustration, we use the following parameters: $\alpha_0 = 0.2$, $\alpha_1 = 0.1$, $\beta_0 = 0.3$, and $\beta_1 = 0.2$. Suppose you observe a sequence generated from this model. For a sufficiently long sequence, a particle filter with resampling will eventually delete most conditionally unlikely particles, due to the fact that there is some probability on each step that any given unlikely particle will be deleted. The particle filter will thus suffer from degeneracy for long sequences. On the other hand, without resampling the approximation will degrade over time because conditionally unlikely particles are never replaced by better particles. For this reason, it is sometimes suggested that resampling only be performed when the effective sample size (ESS) falls below some threshold.

The ESS is calculated as $\text{ESS} = \frac{1}{\sum_{k=1}^{K}(w^k)^2}$. A low ESS means that most of the weight is being placed on a small number of particles, and hence the approximation may be degenerate (although in some cases this may mean that the target distribution is peaky). We evaluated particle filtering with multinomial resampling on synthetic data generated from the HMM described above. Approximation accuracy was measured by using the forward-backward algorithm to compute the hidden state posterior marginals exactly and then comparing these marginals to the particle approximation. Figure 3 shows performance as a function of ESS threshold, demonstrating that there is a fairly narrow range of thresholds for which performance is good. Thus in practice, successful applications of particle filtering may require computationally expensive tuning of this threshold.

In contrast, DPVI achieves performance comparable to the optimal particle filter, but without a tunable threshold. This occurs because DPVI uses an implicit threshold that

is automatically tuned to the problem. Instead of resampling particles, DPVI deletes or propagates particles deterministically based on their relative contribution to the variational bound. We can always incrementally add particles, unlike tuning the threshold. So although K can be viewed as a tuning parameter, we can adapt it with relatively little expense, monotonically increasing the approximation quality in a way that can be easily quantified.

6.2 Dirichlet Process Mixture Model

A DPMM generates data from the following process (Antoniak, 1974; Escobar and West, 1995):

$$G \sim \mathrm{DP}(\alpha, G_0), \qquad \theta_n | G \sim G, \qquad y_n | \theta_n \sim F(\theta_n),$$

where $\alpha \geq 0$ is a concentration parameter and G_0 is a base distribution over the parameter θ_n of the observation distribution $F(y_n|\theta_n)$. Since the Dirichlet process induces clustering of the parameters θ into K distinct values, we can equivalently express this model in terms of a distribution over cluster assignments, $x_n \in \{1, \ldots, C\}$. The distribution over x is given by the Chinese restaurant process (Aldous, 1985):

$$P(x_n = c | x_{1:n-1}) \propto \begin{cases} t_c & \text{if } k \leq C_+ \\ \alpha & \text{if } c = C_+ + 1, \end{cases} \tag{19}$$

where t_c is the number of data points prior to n assigned to cluster c and C_+ is the number of clusters for which $t_c > 0$.

6.2.1 Synthetic Data

We first demonstrate our approach on synthetic data sets drawn from various mixtures of bivariate Gaussians (see Table 1). The model parameters for each simulated data set were chosen to create a spectrum of increasingly overlapping clusters. In particular, we constructed models out of the following building blocks:

$$\mu_1 = \begin{pmatrix} 0.0, & 0.0 \end{pmatrix}, \qquad \mu_2 = \begin{pmatrix} 0.5, & 0.5 \end{pmatrix}$$
$$\Sigma_1 = \begin{pmatrix} 0.25, & 0.0 \\ 0.0, & 0.25 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0.5, & 0.0 \\ 0.0, & 0.5 \end{pmatrix}.$$

For the DPMM, we used a Normal likelihood with a Normal-Inverse-Gamma prior on the component parameters:

$$y_{nd} | x_n = k \sim \mathcal{N}(m_{kd}, \sigma^2_{kd}), \qquad m_{kd} \sim \mathcal{N}(0, \sigma^2_{kd}/\tau), \qquad \sigma^2_{kd} \sim \mathrm{IG}(a, b), \tag{20}$$

where $d \in \{1, 2\}$ indexes observation dimensions and $\mathrm{IG}(a, b)$ denotes the Inverse Gamma distribution with shape a and scale b. We used the following hyperparameter values: $\tau = 25, a = 1, b = 1, \alpha = 0.5$.

Clustering accuracy was measured quantitatively using V-measure (Rosenberg and Hirschberg, 2007). V-measure is an entropy-based measure which explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied. Figure 4 graphically demonstrates the discovery of latent clusters for both DPVI as well as particle filtering. As

Data set	PF ($K = 20$)	MF	DPVI ($K = 1$)	DPVI ($K = 20$)
D1: $[\mu_1, 4\mu_2, 8\mu_2], \Sigma_1$	0.97±0.03	0.80±0.10	0.93±0.05	0.99±0.02
D2: $[\mu_1, 4\mu_2, 8\mu_2], \Sigma_2$	0.89±0.05	0.63±0.02	0.86±0.07	0.90±0.03
D3: $[\mu_1, 2\mu_2, 4\mu_2], \Sigma_1$	0.58±0.12	0.53±0.05	0.51±0.03	0.74±0.16
D4: $[\mu_1, 2\mu_2, 4\mu_2], \Sigma_2$	0.50±0.06	0.29±0.05	0.46±0.05	0.55±0.07
D5: $[\mu_1, \mu_2, 2\mu_2], \Sigma_1$	0.05±0.05	0.28±0.12	0.014±0.02	0.14±0.10
D6: $[\mu_1, \mu_2, 2\mu_2], \Sigma_2$	0.15±0.08	0.12±0.03	0.11±0.06	0.19±0.07

Table 1: Clustering accuracy (V-Measure) for DPMM. Each data set consisted of 200 points drawn from a mixture of 3 Gaussians. For each data set, we repeated the experiment 150 times by iterating through random seeds, reporting mean and standard error. The left column shows the ground truth mean for each cluster and the covariance matrix (shared across clusters). PF denotes particle filtering and MF denotes mean-field.

Number of particles	**DPVI**	**Particle Filtering**
$K = 10$	15.20s	14.71s
$K = 50$	153.75s	184.17s
$K = 100$	567.84s	699.43s

Table 2: Run time comparison for DPMM with synthetic data using data set from Table 1. The run time of DPVI is slightly better than particle filtering for a single pass through the data set.

shown in Table 1, we observe only marginal improvements when the means are farthest from each other and variances are small, as these parameters leads to well-separated clusters in the training set. However, the relative accuracy of DPVI increases considerably when the clusters are overlapping, either due to the fact that the means are close to each other or the variances are high. One factor contributing to greater performance of DPVI might be the diversity term in the variational formulation.

An interesting special case is when $K = 1$. In this case, DPVI is equivalent to the greedy algorithm proposed by Daume (2007) and later extended by Wang and Dunson (2011). In fact, this algorithm was independently proposed in cognitive psychology by Anderson (1991). As shown in Table 1, DPVI with 20 particles outperforms the greedy algorithm, as well as particle filtering with 20 particles. We also demonstrate the run-time performance of DPVI compared to particle filtering in Table 2. It can be seen in our experiments that DPVI tends to be comparable to particle filtering or sometimes more efficient in terms of run-time for the same task, although the theoretical complexity is the same.

6.2.2 Spike Sorting

Spike sorting is an important problem in experimental neuroscience settings where researchers collect large amounts of electrophysiological data from multi-channel tetrodes.

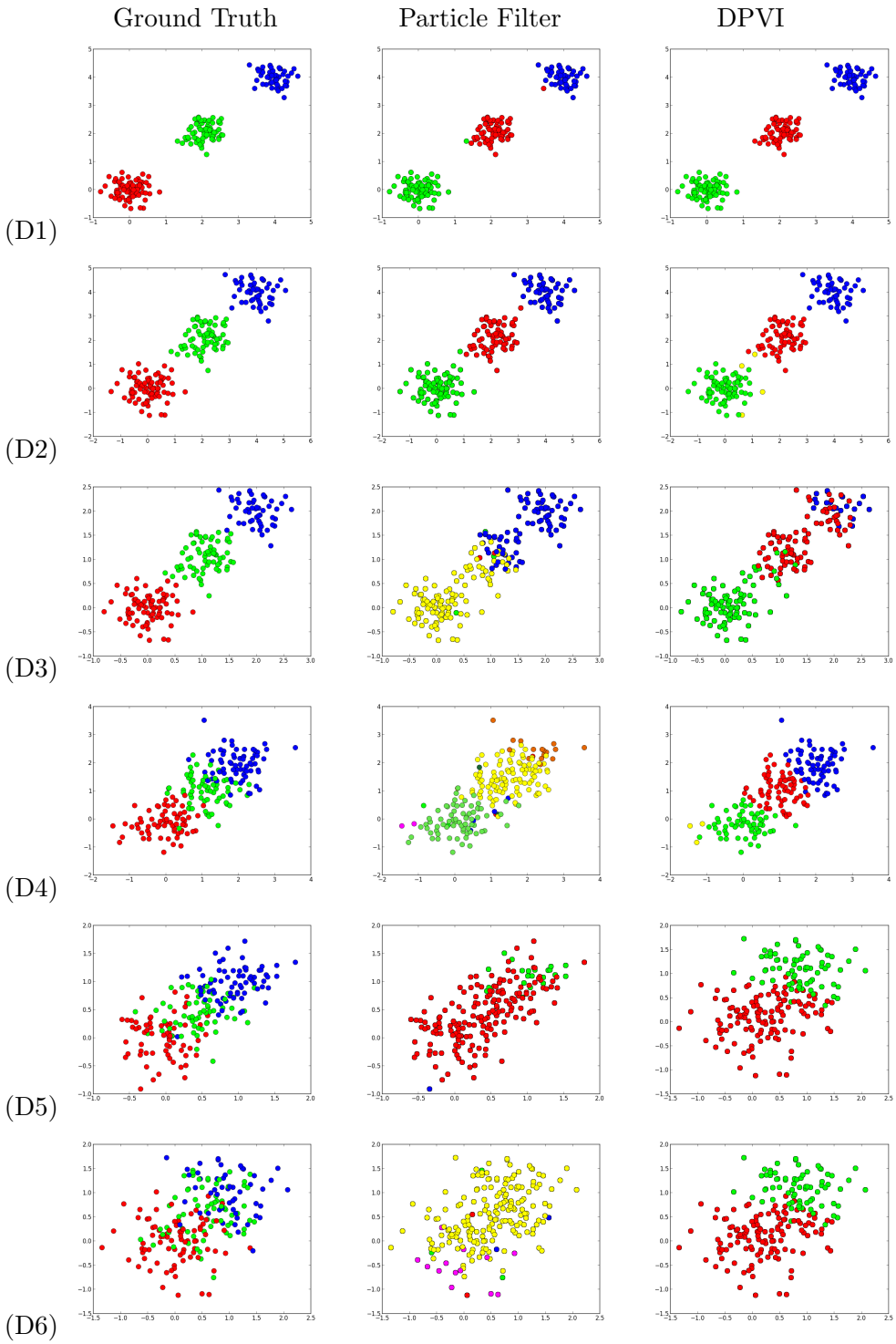


Figure 4: DPMM clustering of synthetic data sets. We treat DPMM as a filtering problem, analyzing one randomly chosen data point at a time. Colors indicate cluster assignments. Each row corresponds to one synthetic data set; refer to Table 1 for corresponding quantitative results. Column 1: Ground truth; Column 2: particle filtering; Column 3: DPVI. The DPVI filter scales similarly to the particle filter but does not underfit as severely.

The goal is to extract from noisy spike recordings attributes such as the number of neurons, and cluster spikes belonging to the same neuron. This problem naturally motivates the use of DPMM, since the number of neurons recorded by a single tetrode is unknown. Previously, Wood and Black (2008) applied the DPMM to spike sorting using particle filtering and Gibbs sampling. Here we show that DPVI can outperform particle filtering, achieving high accuracy even with a small number of particles.

We used data collected from a multiunit recording from a human epileptic patient (Quiroga et al., 2004). The raw spike recordings were preprocessed following the procedure proposed by Quiroga et al. (2004), though we note that our inference algorithm is agnostic to the choice of preprocessing. The original data consist of an input vector with $D = 10$ dimensions and 9196 data points. Following Wood and Black (2008), we used a Normal likelihood with a Normal-Inverse-Wishart prior on the component parameters:

$$\mathbf{y}_n|x_n = k \sim \mathcal{N}(\mathbf{m}_k, \Lambda_k), \qquad \mathbf{m}_k \sim \mathcal{N}(0, \Lambda_k/\tau), \qquad \Lambda_k \sim \mathrm{IW}(\Lambda_0, \nu), \tag{21}$$

where $\mathrm{IW}(\Lambda_0, \nu)$ denotes the Inverse Wishart distribution with degrees of freedom ν and scale matrix Λ_0. We used the following hyperparameter values: $\nu = D + 1, \Lambda_0 = \mathbf{I}, \tau = 0.01, \alpha = 0.1$.

We compared our algorithm to the current best particle filtering baseline, which uses stratified resampling (Wood and Black, 2008; Fearnhead, 2004). The same model parameters were used for all comparisons. Qualitative results, shown in Figure 5B, demonstrate that DPVI is better able to separate the spike waveforms into distinct clusters, despite running DPVI with 10 particles and particle filtering with 100 particles. We also provide quantitative results by calculating the held-out log-likelihood on an independent test set of spike waveforms. The quantitative results (summarized in Table 5C) demonstrate that even with only 10 particles DPVI can outperform particle filtering with 1000 particles.

6.3 Infinite HMM

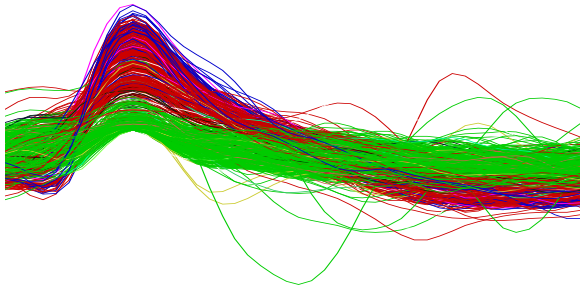
An iHMM generates data from the following process (Teh et al., 2006):

$$\begin{aligned} &G_0 \sim \mathrm{DP}(\gamma, H), \qquad G_k|G_0 \sim \mathrm{DP}(\alpha, G_0), \\ &x_n|x_{n-1} \sim G_{x_{n-1}}, \qquad \theta_k \sim H, \qquad y_n|x_n \sim F(\theta_{x_n}). \end{aligned}$$

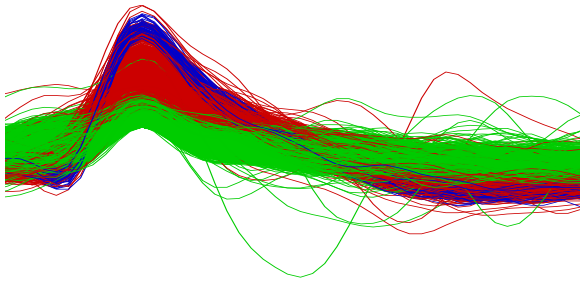
Like the DPMM, the iHMM induces a sequence of cluster assignments. The distribution over cluster assignments is given by the Chinese restaurant franchise (Teh et al., 2006). Letting t_{jc} denote the number of times cluster j transitioned to cluster c, x_n is assigned to cluster c with probability proportional to $t_{x_{n-1}c}$, or to a cluster never visited from x_{n-1} ($t_{x_{n-1}c} = 0$) with probability proportional to α. If an unvisited cluster is selected, x_n is assigned to cluster c with probability proportional to $\sum_j t_{jc}$, or to a new cluster (i.e., one never visited from any state, $\sum_j t_{jc} = 0$) with probability proportional to γ.

6.3.1 Synthetic Data

We generated 50 sequences with length 500 from 50 different HMMs, each with 10 hidden and 5 observed states. For the rows of the transition and initial probability matrices of the



(A)



(B)

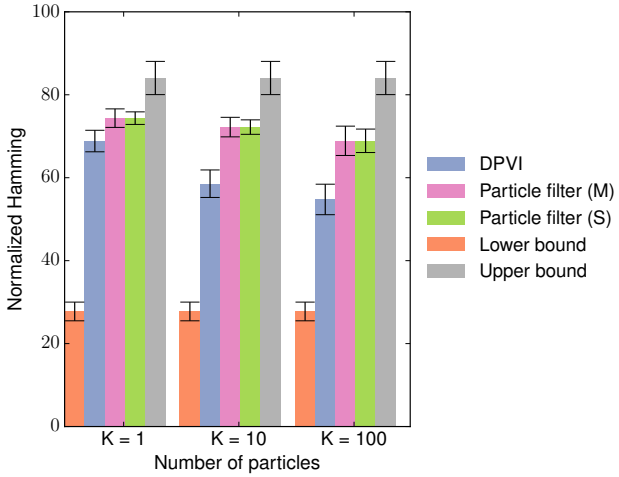
Method	**Predictive LL**
DPVI ($K = 10$)	-3.2474$\times 10^5$ ($\hat{C} = 3$)
DPVI ($K = 100$)	**-1.3888$\times 10^5$** ($\hat{\mathbf{C}} = \mathbf{3}$)
PF ($K = 10$)	-1.4771$\pm$0.21 $\times 10^6$ ($\hat{C} = 37$)
PF ($K = 100$)	-5.6757$\pm$1.14 $\times 10^5$ ($\hat{C} = 13$)
PF ($K = 1000$)	-3.2965$\times 10^5$ ($\hat{C} = 5$)

(C)

Method	**Run time**
DPVI ($K = 10$)	36.20s
DPVI ($K = 50$)	144.6s
DPVI ($K = 100$)	313.8s
PF ($K = 10$)	124s
PF ($K = 50$)	334.2s
PF ($K = 100$)	454.2s

(D)

Figure 5: Spike sorting using the DPMM. Each line is an individual spike waveform, colored according to the inferred cluster. (A) Result using particle filtering with 100 particles and stratified resampling as reported in Wood and Black (2008). (B) Result using DPVI. The same model parameters were used for both particle filtering and DPVI. (C) Spike sorting predictive log-likelihood scores for 200 test points. The best performance is achieved by DPVI with 100 particles. Shown in parentheses is the *maximum a posteriori* number of clusters, $\hat{C}$. (D) Run time comparison for DPMM obtained by using the spike sorting data set. The run time of DPVI is slightly better than particle filtering.



(A)

Method	**Runtime (sec)**
DPVI ($K = 1$)	1.28
DPVI ($K = 10$)	3.56
DPVI ($K = 100$)	204.42
PF ($K = 1$)	1.14
PF ($K = 10$)	1.92
PF ($K = 100$)	31.99

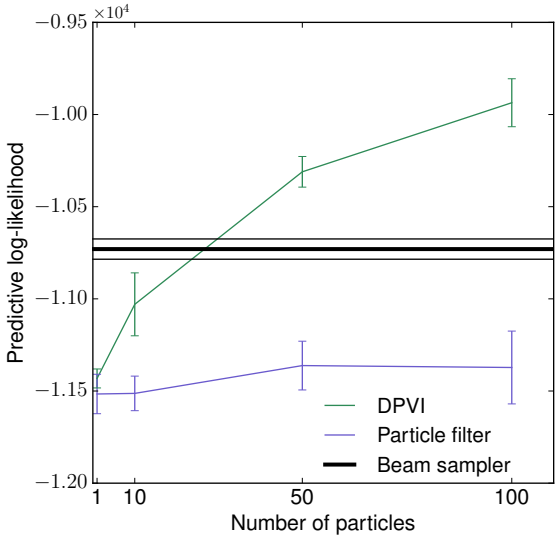
(B)

Figure 6: Infinite HMM on the synthetic data set. (A) Results on 500 synthetic data points generated from an HMM with 10 hidden states. Error is the Hamming distance between the true hidden sequence and the sampled sequence, averaged over 50 data sets. M: multinomial resampling; S: stratified resampling. Lower bound is the expected Hamming distance between data-generating distribution and ground truth. Upper bound is the expected Hamming distance between uniform distribution and ground truth. (B) Run time comparison for the synthetic iHMM data set. We denote particle filtering method by PF.

HMMs we used a symmetric Dirichlet prior with concentration parameter 0.1; for the emission probability matrix, we used a symmetric Dirichlet prior with concentration parameter 10.

Figure 6A illustrates the performance of DPVI and particle filtering (with multinomial and stratified resampling) for varying numbers of particles ($K = 1, 10, 100$). Performance error was quantified by computing the Hamming distance between the true hidden sequence and the sampled sequence. The Munkres algorithm was used to maximize the overlap between the two sequences. The results show that DPVI outperforms particle filtering in all three cases.

When the data consist of long sequences, resampling at every step will produce degeneracy in particle filtering; this tends to result in a smaller number of clusters relative to DPVI. The superior accuracy of DPVI suggests that a larger number of clusters is necessary to capture the latent structure of the data. Not surprisingly, this leads to longer run times (Figurets 6B), but it is important to note that particle filtering and DPVI have comparable per-cluster time complexity.



(A)

Method	**Runtime (sec)**
DPVI ($K = 1$)	4.73
DPVI ($K = 10$)	41.62
DPVI ($K = 100$)	1685
PF ($K = 1$)	1.64
PF ($K = 10$)	28.08
PF ($K = 100$)	211.66
Beam sampler	260

(B)

Figure 7: Infinite HMM for the text analysis task. (A) Predictive log-likelihood for the Alice in Wonderland data set. (B) Results using the "Alice in Wonderland" data set. The maximum number of inferred clusters is $\hat{C} = 147$ for DPVI ($K = 100$) and $\hat{C} = 21$ for particle filter ($K = 100$). As expected, the per cluster runtime of the two methods are comparable.

6.3.2 Text Analysis

We next analyzed a real-world data set, text taken from the beginning of "Alice in Wonderland", with 31 observation symbols (letters). We used the first 1000 characters for training,

and the subsequent 4000 characters for test. Performance was measured by calculating the predictive log-likelihood. We fixed the hyperparameters α and γ to 1 for both DPVI and the particle filtering.

We ran one pass of DPVI (filtering) and particle filtering over the training sequence. We then sampled 50 data sets from the distribution over the sequences. We truncated the number of states and used the learned transition and emission matrices to compute the predictive log-likelihood of the test sequence. To handle the unobserved emissions in the test sequence we used "add-δ" smoothing with $\delta = 1$. Finally, we averaged over all the 50 data sets.

We also compared DPVI to the beam sampler (Van Gael et al., 2008), a combination of dynamic programming and slice sampling, which was previously applied to this data set. For the beam sampler, we followed the setting of Van Gael et al. (2008). We run the sampler for 10000 iterations and collect a sample of hidden state sequence every 200 iterations. Figure 6B shows the predictive log-likelihood for varying numbers of particles. Even with a small number of particles, DPVI can outperform both particle filtering and the beam sampler.

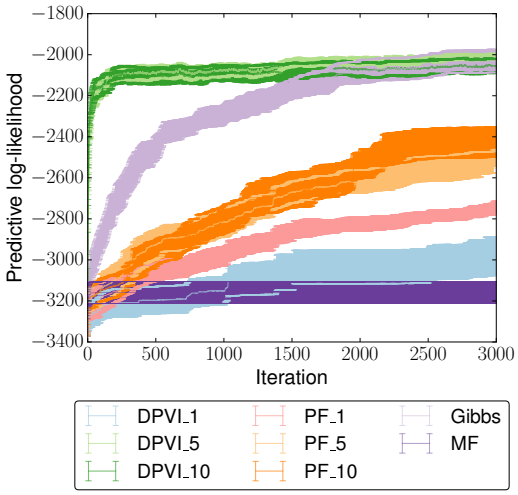
6.3.3 User Behavior Analysis

We analyzed a data set of user behavior in a photo editing software application. The data set contains sequences of edits applied by users to different photos. An iHMM can be utilized for better understanding the high-level tasks of the users. That is, a sequence of multiple edits may be required to perform a high-level task such as cropping or masking a photo. Our data set contains 30000 edits from which we use 1000 data points as a held-out set. There are 23 possible observations (edits) in the data set. We compare DPVI with 1, 5 and 10 particles with particle filtering, Gibbs sampling and mean field inference schemes. For all the inference schemes, we set the hyperparameters α and γ to 1. In every iteration of DPVI and particle filtering, we do a forward filtering-backward smoothing pass over all the sequences of the data set. Our results, shown in Figure 8(B), demonstrate that DPVI with 5 and 10 particles can converge in fewer iterations compared to other reasonable baselines. We illustrate the lower bound (Eq. 10) convergence in Figure 9. The results are computed over 20 runs and for 1, 5 and 10 particles.

It is important to note here that we do not expect DPVI to outperform Gibbs sampling in all scenarios; when computation time is not strongly limited, we expect DPVI and Gibbs to perform similarly. This point applies here as well as to the experiments reported in the sections below. We see DPVI as a useful alternative to Gibbs when the computational budget is low and the required fidelity of the approximation can be satisfied by capturing a few of the posterior modes.

6.4 Infinite Relational Model (IRM)

The IRM (Kemp et al., 2006) is a nonparametric model of relational systems. The model simultaneously discovers the clusters of entities and the relationships between the clusters. A key assumption of the model is that each entity belongs to exactly one cluster.



(A)

Method	**Predictive LL**
DPVI ($K = 1$)	-2983.24 ± 93.77
DPVI ($K = 5$)	$\mathbf{-2018.02 \pm 9.23}$
DPVI ($K = 10$)	$\mathbf{-2058.75 \pm 33.62}$
PF ($K = 1$)	-2743.60 ± 28.49
PF ($K = 5$)	-2503.95 ± 40.46
PF ($K = 10$)	-2424.01 ± 72.72
Mean field	-3158.69 ± 52.67
Gibbs	$\mathbf{-2030.50 \pm 56.55}$

(B)

Figure 8: Infinite HMM results for the user behavior analysis task. (A) Predictive log-likelihood vs iteration for the user behavior data set. The error bars correspond to the standard error and are computed over 20 runs. In every iteration of DPVI and particle filtering, we do a forward filtering-backward smoothing pass over all the sequences of the data set. (B) Predictive log-likelihood after 3000 iterations of DPVI, Gibbs, particle filter and mean field for the user behavior data set (with 1000 data points as held-out). The best performance is achieved by DPVI with 5-10 particles and also the Gibbs sampler. Run time for each epoch is as follows: 0.726 sec for mean field, 0.593 sec for Gibbs, 4.52 sec for particle filter with 10 particles, and 4.74 sec for DPVI with 10 particles.

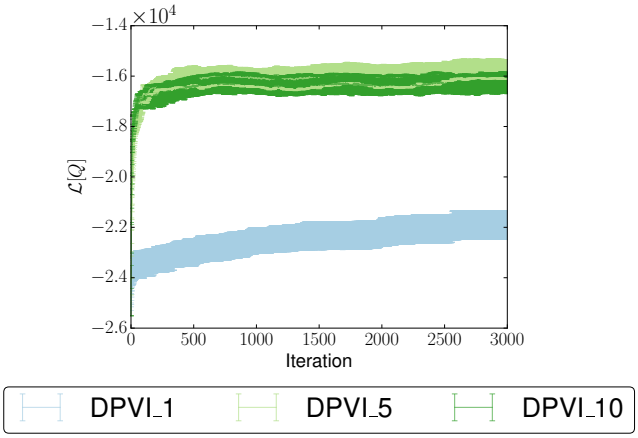


Figure 9: Infinite HMM results for the user behavior analysis task. Lower bound vs iteration for the user behavior data set. The error bars correspond to the standard error and are computed over 20 runs. In every iteration of DPVI, we do a forward filtering-backward smoothing pass over all the sequences of the data set.

Given a relation R involving J types of entities, the goal is to infer a vector of cluster assignments x^j for all the entities of each type $j = 1, \ldots, J$.[2] Assuming the cluster assignments for each type are independent, the joint density of the relation and the cluster assignment vectors can be written as:

$$P(R, x^1, \ldots, x^J) = P(R|x^1, \ldots, x^J) \prod_{j=1}^{J} P(x^j). \tag{22}$$

The cluster assignment vectors are drawn from a CRP(α) prior. Given the cluster assignment vectors, the relations are drawn from a Bernoulli distribution with a parameter η that depends on the clusters involved in that relation. For instance, in a single two-place relation, $\eta(a, b)$ is the probability of having a link between any given pair (i, j) where i is in cluster a and j is in cluster b.

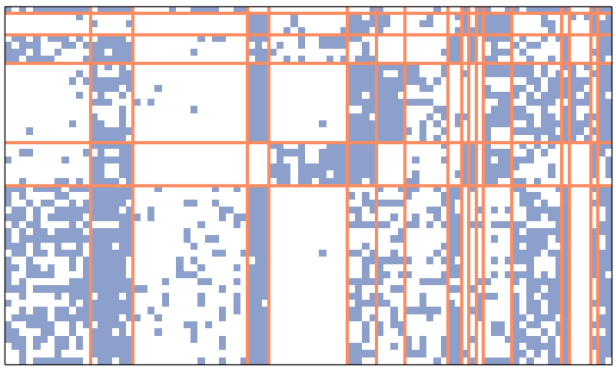
More formally, let us define an M dimensional relation $R : T^{d_1} \times \ldots T^{d_M} \mapsto \{0, 1\}$, over J different types. Each relational value is generated according to:

$$R(i_1, \ldots, i_M)|x^1, \ldots, x^J \sim \text{Bernoulli}(\eta(x_{i_1}^{d_1}, \ldots, x_{i_M}^{d_M})), \tag{23}$$

where d_m denotes the label of the type (i.e., $d_m \in \{1, \cdots, J\}$) and i_m is the entity occupying position m in the relation. Each entry of parameter matrix η is drawn from a Beta(β, β) distribution. By using a conjugate Beta-Bernoulli model, we can analytically marginalize the parameters η (see Kemp et al., 2006), allowing us to directly compute the likelihood of the relational matrix given the cluster assignments, $P(R|x^1, \ldots, x^J)$.

We compared the performance of DPVI with Gibbs sampling, using predictive log-likelihood on held-out data as a performance metric. The "animals" data set analyzed in

2. The IRM model can be defined for multiple relations but for simplicity we only describe the single relation case.



Sample animal clusters:
A1: Hippopotamus, Elephant, Rhinoceros
A2: Seal, Walrus, Dolphins, Blue Whale, Killer Whale, Humpback Whale
A3: Beaver, Otter, Polar Bear

Sample feature clusters:
F1: Hooves, Long neck, Horns
F2: Inactive, Slow, Bulbous Body, Tough Skin
F3: Lives in Fields, Lives in Plains, Grazer
F4: Walks, Quadrupedal, Ground
F5: Fast, Agility, Active, Tail

Figure 10: Co-clustering of animals (rows) and features (columns) after 50 iterations of DPVI with 10 particles in the infinite relational model.

Kemp et al. (2006), was used for this task. This data set (Osherson et al., 1991) is a two type data set $R : T_1 \times T_2 \to \{0, 1\}$ with animals and features as it types; it contains 50 animals and 85 features.

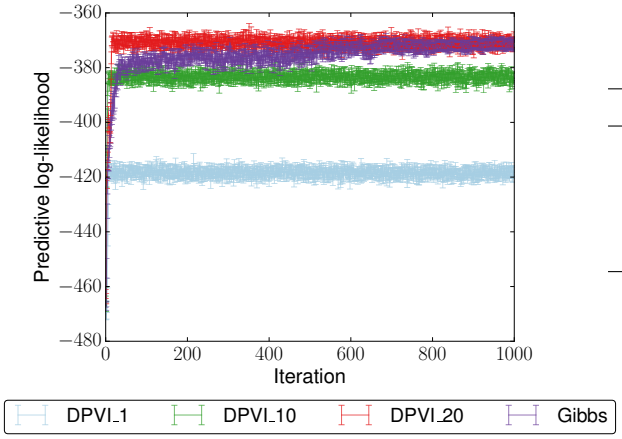
We removed 20% of the relations from the data set and computed the predictive log-likelihood for the held-out data. We ran DPVI with 1, 10 and 20 particles for 1000 iterations. Given the weights of the particles, we computed the weighted log-likelihood. We also ran 20 independent runs of the Gibbs sampler and DPVI for 1000 iterations and computed the average predictive log-likelihood. Every iteration scans all the data points in all the types sequentially. We set the hyperparameters α and β to 1. Figure 10 illustrates the co-clustering discovered by DPVI for the data set, demonstrating intuitively reasonable animal and feature clusters.

The results after 1000 iterations are presented in Table 11B. The best performance is achieved by DPVI with 20 particles. Figure 11A shows the predictive log-likelihood for every iteration of DPVI and Gibbs sampling. DPVI with 10 and 20 particles converge in 11 and 18 iterations, respectively. In terms of computation time per iteration of DPVI versus Gibbs, the only difference for DPVI with one particle and Gibbs is the sorting cost. Hence, for the multiple particle versus multiple runs of Gibbs sampling, the only additional cost is the sorting cost for multiple particles (e.g. 10 or 20). However, this insignificant additional cost is compensated for by a faster convergence rate in our experiments.

6.5 Ising Model

So far, we have been studying inference in directed graphical models, but DPVI can also be applied to undirected graphical models. We illustrate this using the Ising model for binary vectors $x \in \{-1, +1\}^N$:

$$f(x) = \exp\left\{\frac{1}{2} x W x^\top + \theta x^\top\right\}, \tag{24}$$



(A)

Method	Predictive LL
DPVI ($K = 1$)	-418.49 ± 4.24
DPVI ($K = 10$)	-382.46 ± 7.34
DPVI ($K = 20$)	$\mathbf{-370.67 \pm 6.23}$
Gibbs (avg. of 20 runs)	$\mathbf{-371.74 \pm 5.07}$

(B)

Figure 11: Infinite relational model results for the animals data set. (A) Predictive log-likelihood vs iteration for the animals data set. The error bars correspond to the standard error and for both methods are computed over 20 runs. (B) Predictive log-likelihood after 1000 iterations of DPVI and Gibbs (with 50 burnin iterations) for the animals data set (with 20 % held-out). DPVI with 20 particles performs in par with the Gibbs sampler.

where $W \in \mathbb{R}^{N \times N}$ and $\theta \in \mathbb{R}^N$ are fixed parameters. In particular, we study a square lattice ferromagnet, where $W_{ij} = \beta$ for neighboring nodes (0 otherwise) and $\theta_i = 0$ for all nodes. We refer to β as the *coupling strength*. This model has two global modes: when all the nodes are set to 1, and when all the nodes are set to 0. As the coupling strength increases, the probability mass becomes increasingly concentrated at the two modes.

We applied DPVI to this model, varying the number of particles and the coupling strength. At each iteration, we evaluated the change in log probability that would result from setting $\tilde{x}_n^k = 1$:

$$a_n^k = \sum_{n' \in c_n} W_{n,n'} x_{n'}^k + \theta_n, \tag{25}$$

and likewise the change for setting $\tilde{x}_n^k = 1$ can be computing by simply flipping the sign of a_n^k. Ordering these changes, we then took the top K to determine the new particle set.

To quantify performance, we computed the DPVI variational lower bound on the partition function and compared this to the lower bound furnished by the mean-field approximation (see Wainwright and Jordan, 2008). Figure 12A shows the results of this analysis for low coupling strength ($\beta = 0.01$) and high coupling strength ($\beta = 100$). DPVI consistently achieves a better lower bound than mean-field, even with a single particle, and this advantage is especially conspicuous for high coupling strength. Adding more particles improves the results, but more than 3 particles does not appear to confer any additional improvement for high coupling strength. These results illustrate how DPVI is able to capture multimodal target distributions, where mean-field approximations break down (since they cannot effectively handle multimodality).

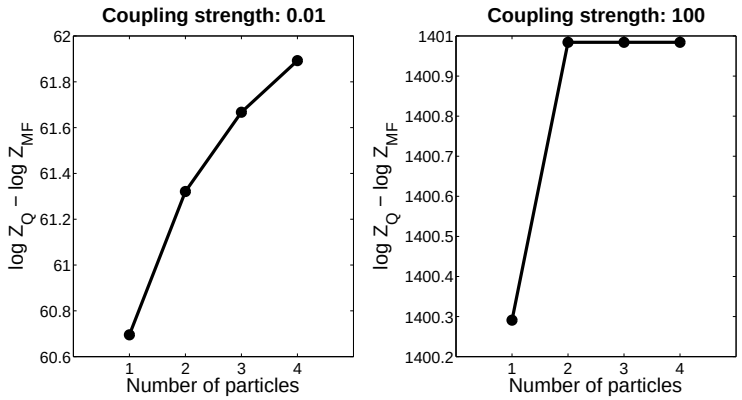


Figure 12: Ising model results. Difference between DPVI and mean-field lower bounds on the partition function. Positive values indicates superior DPVI performance. (A) Low coupling strength; (B) high coupling strength.

To illustrate the performance of DPVI further, we compared several posterior approximations for the Ising model in Figure 13. In addition to the mean-field approximation, we also compared DPVI with two other standard approximations: the Swendsen-Wang Monte Carlo sampler (Swendsen and Wang, 1987) and loopy belief propagation (Murphy et al., 1999). The sampler tended to produce noisy results, whereas mean-field and BP both failed to capture the multimodal structure of the posterior. In contrast, DPVI with two particles perfectly captured the two modes.

7. Conclusions

This paper introduced a particle-based variational method that applies to a broad class of inference problems in discrete models. We described a practical algorithm for optimizing the particle approximation, and showed empirically that it can outperform widely-used Monte Carlo and variational algorithms. The key to the success of this approach is an intentional selection of particles: rather than generating them randomly (as in Monte Carlo algorithms), we deterministically choose a set of unique particles that optimizes the KL divergence between the approximation and the target distribution.

This approach leads to an interesting view on the problem of resampling in sequential Monte Carlo. Resampling is necessary to remove conditionally unlikely particles, but the resulting loss of particle diversity can lead to degeneracy. As we showed in our experiments, tuning an ESS threshold for resampling can improve performance, but requires finding a relatively narrow sweet spot for the threshold. DPVI achieves comparable performance to the best particle filter by using a deterministic strategy for deleting and replacing particles and does not require tuning thresholds. Each particle is guaranteed to be unique and have high probability among all states discovered so far.

DPVI also suggests new hybrids of ideas from Monte Carlo and variational inference. Consider models where all particle extensions cannot be enumerated. In this setting, one could randomly choose particle extensions. To use these particles in a Monte Carlo scheme

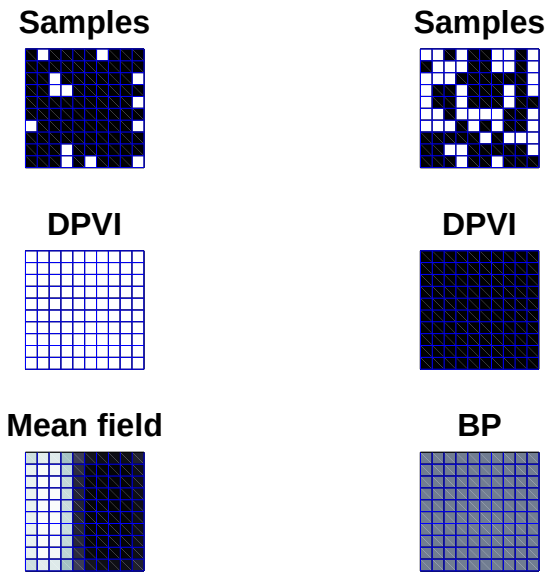


Figure 13: Ising model simulations. Examples of posteriors for the ferromagnetic lattice at low coupling strength. (Top) Two configurations from a Swendsen-Wang sampler. (Middle) Two DPVI particles. (Bottom left) Mean-field expected value. (Bottom right) Loopy belief propagation expected value.

we may need to know the output probability density of the particle extension mechanism. However, if we use the results in DPVI, we just need to be able to score the results under the joint probability distribution. No proposal distribution is needed. This could make it possible to use proposal mechanisms that can be seen to work well empirically but that are difficult to analyze *a priori*.

Although our empirical results are promising, much more empirical and theoretical work is needed to understand the fundamental tradeoffs between variational and Monte Carlo inference. However, the results for DPVI on several problems are promising, and the approach to defining variational approximations may be more broadly applicable. We hope this work encourages others to develop different hybrids of Monte Carlo and variational inference that overcome the limitations of each approach when used in isolation.

Acknowledgments

TDK was generously supported by the Leventhal Fellowship. VKM is supported by the Army Research Office Contract Number 0010363131, Office of Naval Research Award N000141310333, and the DARPA PPAML program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government. The U.S. Government is

References

David Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII1983*, pages 1–198, 1985.

John R Anderson. The adaptive nature of human categorization. *Psychological Review*, 98: 409–429, 1991.

Charles E Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, pages 1152–1174, 1974.

Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*, pages 577–584, 2002.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48:259–302, 1986.

Alexandre Bouchard-Côté and Michael I Jordan. Optimization of structured mean field objectives. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 67–74. AUAI Press, 2009.

Hal Daume. Fast search for Dirichlet process mixture models. In *International Conference on Artificial Intelligence and Statistics*, pages 83–90, 2007.

Arnaud Doucet, Nando De Freitas, Neil Gordon, et al. *Sequential Monte Carlo Methods in Practice*. Springer New York, 2001.

Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

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

Natalia Flerova, Emma Rollon, and Rina Dechter. Bucket and mini-bucket schemes for m best solutions over graphical models. In *Graph Structures for Knowledge Representation and Reasoning*, pages 91–118. Springer, 2012.

Andrew Frank, Padhraic Smyth, and Alexander T Ihler. Particle-based variational inference for continuous systems. In *Advances in Neural Information Processing Systems*, 2009.

Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

Peter Hall, John T Ormerod, and MP Wand. Theory of Gaussian variational approximation for a Poisson mixed model. *Statistica Sinica*, 21:369–389, 2011.

Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17:295–311, 2008.

Tommi S Jaakkola and Michael I Jordan. Improving the mean field approximation via the use of mixture distributions. In MI Jordan, editor, *Learning in Graphical Models*, pages 163–173. Springer, 1998.

Beatrix Jones, Carlos Carvalho, Adrian Dobra, Chris Hans, Chris Carter, and Mike West. Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science*, 20:388–400, 2005.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

Neil D Lawrence. *Variational inference in probabilistic models*. PhD thesis, University of Cambridge, 2000.

Kerrie L Mengersen, Richard L Tweedie, et al. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24:101–121, 1996.

Sean P Meyn and Richard L Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.

Andrew C Miller, Nicholas Foti, and Ryan P Adams. Variational boosting: Iteratively refining posterior approximations. *arXiv preprint arXiv:1611.06585*, 2016.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

Daniel N Osherson, Joshua Stern, Ormond Wilkie, Michael Stob, and Edward E Smith. Default probability. *Cognitive Science*, 15:251–269, 1991.

Rodrigo Quian Quiroga, Zoltan Nadasdy, and Yoram Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural computation*, 16:1661–1687, 2004.

Christian P Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420, 2007.

Philip Schniter, Lee C Potter, and Justin Ziniel. Fast Bayesian matching pursuit. In *Information Theory and Applications Workshop*, pages 326–333. IEEE, 2008.

Robert H Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58:86–88, 1987.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095. ACM, 2008.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

Bo Wang, DM Titterington, et al. Convergence properties of a general algorithm for calculating variational bayesian estimates for a normal mixture model. *Bayesian Analysis*, 1: 625–650, 2006.

Lianming Wang and David B Dunson. Fast Bayesian inference in Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 20, 2011.

Frank Wood and Michael J Black. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173:1–12, 2008.

Chen Yanover and Yair Weiss. Finding the M most probable configurations in arbitrary graphical models. In *Advances in Neural Information Processing Systems*, page None, 2003.