Collapsed Variational Dirichlet Process Mixture Models[*]

Kenichi Kurihara
Dept. of Computer Science
Tokyo Institute of Technology, Japan
kurihara@mi.cs.titech.ac.jp

Max Welling
Dept. of Computer Science
UC Irvine, USA
welling@ics.uci.edu

Yee Whye Teh
Dept. of Computer Science
National University of Singapore
tehyw@comp.nus.edu.sg

Abstract

Nonparametric Bayesian mixture models, in particular Dirichlet process (DP) mixture models, have shown great promise for density estimation and data clustering. Given the size of today's datasets, computational efficiency becomes an essential ingredient in the applicability of these techniques to real world data. We study and experimentally compare a number of variational Bayesian (VB) approximations to the DP mixture model. In particular we consider the standard VB approximation where parameters are assumed to be independent from cluster assignment variables, and a novel collapsed VB approximation where mixture weights are marginalized out. For both VB approximations we consider two different ways to approximate the DP, by truncating the stick-breaking construction, and by using a finite mixture model with a symmetric Dirichlet prior.

1 Introduction

Mixture modeling remains one of the most useful tools in statistics, machine learning and data mining for applications involving density estimation or clustering. One of the most prominent recent developments in this field is the application of nonparametric Bayesian techniques to mixture modeling, which allow for the automatic determination of an appropriate number of mixture components. Current inference algorithms for such models are mostly based on Gibbs sampling, which suffer from a number of drawbacks. Most importantly, Gibbs sampling is not efficient enough to scale up to the large scale problems we face in modern-day data mining. Secondly, sampling requires careful monitoring of the convergence of the Markov chain, both to decide on the number of samples to be ignored for burn-in and to decide how many samples are needed to reduce the variance in the estimates. These considerations have lead researchers to develop deterministic alternatives which trade off variance with bias and are easily monitored in terms of their convergence. Moreover, they can be orders of magnitude faster than sampling, especially when special data structures such as KD trees are used to cache certain sufficient statistics [Moore, 1998; Verbeek *et al.*, 2003; Kurihara *et al.*, 2006].

[Blei and Jordan, 2005] recently applied the framework of variational Bayesian (VB) inference to Dirichlet process (DP) mixture models and demonstrated significant computational gains. Their model was formulated entirely in the truncated stick-breaking representation. The choice of this representation has both advantages and disadvantages. For instance, it is very easy to generalize beyond the DP prior and use much more flexible priors in this representation. On the flip side, the model is formulated in the space of explicit, non-exchangeable cluster labels (instead of partitions). In other words, randomly permuting the labels changes the probability of the data. This then requires samplers to mix over cluster labels to avoid bias [Porteous *et al.*, 2006].

In this paper we propose and study alternative approaches to VB inference in DP mixture models beyond that proposed in [Blei and Jordan, 2005]. There are three distinct contributions in this paper: in proposing an improved VB algorithm based on integrating out mixture weights, in comparing the stick-breaking representation against the finite symmetric Dirichlet approximation to the DP, and in the maintaining optimal ordering of cluster labels in the stick-breaking VB algorithms. These lead to a total of six different algorithms, including the one proposed in [Blei and Jordan, 2005]. We experimentally evaluate these six algorithms and compare against Gibbs sampling.

In Section 2.1 we explore both the truncated stick-breaking approximation and the finite symmetric Dirichlet prior as finite dimensional approximations to the DP. As opposed to the truncated stick-breaking approximation, the finite symmetric Dirichlet model is exchangeable over cluster labels. Theoretically this has important consequences, for example a Gibbs sampler is not required to mix over cluster labels if we are computing averages over quantities invariant to cluster label permutations (as is typically the case).

In Section 2.2 we explore the idea of integrating out the mixture weights $\boldsymbol{\pi}$, hence collapsing the model to a lower dimensional space. This idea has been shown to work well for LDA models [Teh *et al.*, 2006] where strong dependencies exist between model parameters and assignment variables. Such dependencies exist between mixture weights and assignment

[*] This material is based in part upon work supported by the National Science Foundation under Grant Number D/HS-0535278. YWT thanks the Lee Kuan Yew Endowment Fund for funding.

variables in our mixture model context as well, thus collapsing could also be important here. This intuition is reflected in the observation that the variational bound on the log evidence is guaranteed to improve.

In Section 3 we derive the VB update equations corresponding to the approximations in Section 2. We also consider optimally reordering cluster labels in the stick-breaking VB algorithms. As mentioned, the ordering of the cluster labels is important for models formulated in the stick-breaking representation. In the paper [Blei and Jordan, 2005] this issue was ignored. Here we also study the effect of cluster reordering on relevant performance measures such as the predictive log evidence.

The above considerations lead us to six VB inference methods, which we evaluate in Section 4. The methods are: 1) the truncated stick-breaking representation with standard VB (TSB), 2) the truncated stick-breaking representation with collapsed VB (CTSB), 3) the finite symmetric Dirichlet representation with standard VB (FSD), 4) the finite symmetric Dirichlet presentation with collapsed VB (CFSD), and 5) and 6) being TSB and CTSB with optimal reordering (O-TSB and O-CTSB respectively).

2 Four Approximations to the DP

We describe four approximations to the DP in this section. These four approximations are obtained by a combination of truncated stick-breaking/finite symmetric Dirichlet approximations and whether the mixture weights are marginalized out or not. Based on these approximations we describe the six VB inference algorithms in the next section.

The most natural representation of DPs is using the Chinese restaurant process, which is formulated in the space of partitions. Partitions are groupings of the data independent of cluster labels, where each data-point is assigned to exactly 1 group. This space of partitions turns out to be problematic for VB inference, where we wish to use fully factorized variational distributions on the assignment variables, $Q(\mathbf{z}) = \prod_n q(z_n)$. Since the assignments $z_1 = 1, z_2 = 1, z_3 = 2$ represent the same partition $(1,2)(3)$ as $z_1 = 3, z_2 = 3, z_3 = 2$, there are intricate dependencies between the assignment variables and it does not make sense to use the factorization above. We can circumvent this by using finite dimensional approximations for the DP, which are formulated in the space of cluster labels (not partitions) and which are known to closely approximate the DP prior as the number of explicitly maintained clusters grows [Ishwaran and James, 2001; Ishwaran and Zarepour, 2002]. These finite approximations are what will we discuss next.

2.1 TSB and FSD Approximations

In the first approximation we use the stick-breaking representation for the DP [Ishwaran and James, 2001] and truncate it after T terms,

$$v_i \sim \mathcal{B}(v_i; 1, \alpha) \qquad i = 1, ..., T-1 \quad (1)$$

$$v_T = 1 \quad (2)$$

$$\pi_i = v_i \prod_{j<i} (1 - v_j) \qquad i = 1, ..., T \quad (3)$$

$$\pi_i = 0 \qquad i > T \quad (4)$$

where $\mathcal{B}(v; 1, \alpha)$ is a beta density for variable v with parameters 1 and α, and one can verify that $\sum_{i=1}^{T} \pi_i = 1$. Incorporating this into a joint probability over data items $X = \{\mathbf{x}_n\},\ n = 1, ..., N$, cluster assignments $\mathbf{z} = \{z_n\},\ n = 1, ..., N$, stick-breaking weights $\mathbf{v} = \{v_i\},\ i = 1, ..., T$ and cluster parameters $\boldsymbol{\eta} = \{\eta_i\},\ i = 1, ..., T$ we find

$$P(X, \mathbf{z}, \mathbf{v}, \boldsymbol{\eta}) = \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n})\, p(z_n|\boldsymbol{\pi}(\mathbf{v}))\right] \left[\prod_{i=1}^{T} p(\eta_i) \mathcal{B}(v_i; 1, \alpha)\right] \quad (5)$$

where $\boldsymbol{\pi}(\mathbf{v})$ are the mixture weights as defined in (3). In this representation the cluster labels are not interchangeable, i.e. changing labels will change the probability value in (5). Note also that as $T \to \infty$ the approximation becomes exact.

A second approach to approximate the DP is by assuming a finite (but large) number of clusters, K, and using a symmetric Dirichlet prior $\mathcal{D}$ on $\boldsymbol{\pi}$ [Ishwaran and Zarepour, 2002],

$$\boldsymbol{\pi} \sim \mathcal{D}(\boldsymbol{\pi}; \tfrac{\alpha}{K}, ..., \tfrac{\alpha}{K}) \quad (6)$$

This results in the joint model,

$$P(X, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n})\, p(z_n|\boldsymbol{\pi})\right] \left[\prod_{i=1}^{K} p(\eta_i)\right] \mathcal{D}(\boldsymbol{\pi}; \tfrac{\alpha}{K}, ..., \tfrac{\alpha}{K}) \quad (7)$$

The essential difference with the stick-breaking representation is that the cluster labels remain interchangeable under this representation, i.e. changing cluster labels does *not* change the probability [Porteous *et al.*, 2006]. The limit $K \to \infty$ is somewhat tricky because in the transition $K \to \infty$ we switch to the space of partitions, where states that result from cluster relabelings are mapped to the same partition. For example, both $z_1 = 1, z_2 = 1, z_3 = 2$ and $z_1 = 3, z_2 = 3, z_3 = 2$ are mapped to the same partition $(1,2)(3)$.

In figure 1 we show the prior average cluster sizes under the truncated stick-breaking (TSB) representation (left) and under the finite symmetric Dirichlet (FSD) prior (middle) for two values of the truncation level and number of clusters respectively. From this figure it is apparent that the cluster labels in the TSB prior are not interchangeable (the probabilities are ordered in decreasing size), while they are interchangeable for the FSD prior. As we increase T and K these priors approximate the DP prior with increasing accuracy.

One should note however, that they live in different spaces. The DP itself is most naturally defined in the space of partitions, while both TSB and FSD are defined in the space over cluster labels. However, TSB and FSD also live in different

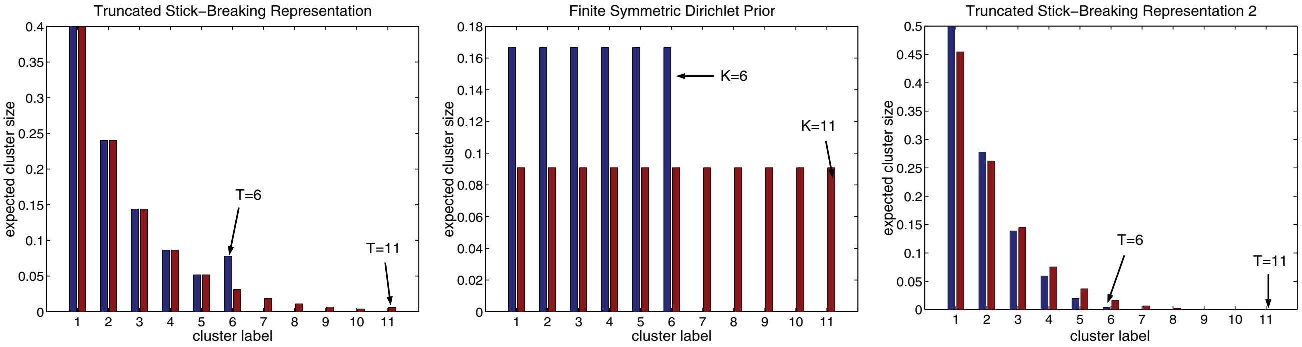

Figure 1: Average cluster size for three finite approximations to the DP prior. Left: Truncated stick-breaking prior (TSB) as given in (3). Middle: Finite Symmetric Dirichlet prior (FSD). Right: Stick-breaking representation corresponding to the FSD prior. In each figure we show results for two truncation levels: $T/K = 6$ (left bars) and $T/K = 11$ (right bars).

spaces! More precisely, one can transform a sample from the FSD prior into the stick-breaking representation by performing a size-biased permutation of the mixture weights $\boldsymbol{\pi}$ (i.e. after every sample from $\mathcal{D}(\boldsymbol{\pi})$ we sample an ordering according to $\boldsymbol{\pi}$ without replacement). As it turns out, for finite K this does not exactly recover the left hand figure in 1, but rather samples from a prior very closely related to it shown in the right pane of figure 1. This prior is given by a stick-breaking construction as in eqn.(3) with stick-lengths sampled from,

$$v_i \sim \mathcal{B}(v_i; 1 + \frac{\alpha}{K}, \alpha - \frac{i\alpha}{K}) \tag{8}$$

Conversely, we can obtain samples from the FSD prior by applying a random, uniformly distributed permutation on the cluster weights obtained from eqn.(8). Although these two stick-breaking constructions are slightly different, for large enough K, T they are very similar and we do not expect any difference in terms of performance between the two.

2.2 Marginalizing out the Mixture Weights

The variational Bayesian approximations discussed in the next section assume a factorized form for the posterior distribution. This means that we assume that parameters are independent of assignment variables. This is clearly a very bad assumption because changes in $\boldsymbol{\pi}$ will have a considerable impact on $\mathbf{z}$. Ideally, we would integrate out all the parameters, but this is too computationally expensive. There is however a middle ground: we can marginalize out $\boldsymbol{\pi}$ from both methods without computational penalty if we make another approximation which will be discussed in section 3.3. For both TSB and FSD representations the joint collapsed model over $X, \mathbf{z}, \boldsymbol{\eta}$ is given by,

$$P(X, \mathbf{z}, \boldsymbol{\eta}) = \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n})\right] p(\mathbf{z}) \left[\prod_{i=1}^{\infty} p(\eta_i)\right] \tag{9}$$

with different distributions over cluster labels $p(\mathbf{z})$ in both cases. For the TSB representation we have,

$$p_{\text{TSB}}(\mathbf{z}) = \prod_{i<T} \frac{\Gamma(1+N_i)\Gamma(\alpha + N_{>i})}{\Gamma(1+\alpha+N_{\geq i})} \tag{10}$$

with

$$N_i = \sum_{n=1}^{N} \mathbb{I}(z_n = i) \qquad N_{>i} = \sum_{n=1}^{N} \mathbb{I}(z_n > i) \tag{11}$$

and $N_{\geq i} = N_i + N_{>i}$. For FSD we find instead,

$$p_{\text{FSD}}(\mathbf{z}) = \frac{\Gamma(\alpha)\prod_{k=1}^{K}\Gamma(N_k + \frac{\alpha}{K})}{\Gamma(N+\alpha)\Gamma(\frac{\alpha}{K})^K} \tag{12}$$

3 Variational Bayesian Inference

The variational Bayesian inference algorithm [Attias, 2000; Ghahramani and Beal, 2000] lower bounds the log marginal likelihood by assuming that parameters and hidden variables are independent. The lower bound is given by,

$$\mathcal{L}(X) \geq \mathbf{B}(X) = \sum_{\mathbf{z}} \int_{\text{d}\boldsymbol{\theta}} Q(\mathbf{z})Q(\boldsymbol{\theta}) \log \frac{P(X, \mathbf{z}, \boldsymbol{\theta})}{Q(\mathbf{z})Q(\boldsymbol{\theta})} \tag{13}$$

where $\boldsymbol{\theta}$ is either $\{\boldsymbol{\eta}, \mathbf{v}\}$, $\{\boldsymbol{\eta}, \boldsymbol{\pi}\}$ or $\{\boldsymbol{\eta}\}$ in the various DP approximations discussed in the previous section. Approximate inference is then achieved by alternating optimization of this bound over $Q(\mathbf{z})$ and $Q(\boldsymbol{\theta})$. In the following we will spell out the details of VB inference for the proposed four methods. For the TSB prior we use,

$$Q_{\text{TSB}}(\mathbf{z}, \boldsymbol{\eta}, \mathbf{v}) = \left[\prod_{n}^{N} q(z_n)\right] \left[\prod_{i=1}^{T} q(\eta_i) q(v_i)\right] \tag{14}$$

where $q(\mathbf{v})$ is not used in the TSB model with $\mathbf{v}$ marginalized out. For the FSD prior we use,

$$Q_{\text{FSD}}(\mathbf{z}, \boldsymbol{\eta}, \boldsymbol{\pi}) = \left[\prod_{n}^{N} q(z_n)\right] \left[\prod_{k=1}^{K} q(\eta_k)\right] q(\boldsymbol{\pi}) \tag{15}$$

As well, $q(\boldsymbol{\pi})$ is left out for the collapsed version.

3.1 Bounds on the Evidence

Given the variational posteriors we can construct bounds on the log marginal likelihood by inserting Q into eqn.(13). After some algebra we find the following general form,

$$
\begin{aligned}
\mathbf{B}(X) = &\sum_{n=1}^{N} \sum_{z_n} \int_{\mathrm{d}\eta_{z_n}} q(z_n) q(\eta_{z_n}) \log p(\mathbf{x}_n|\eta_{z_n}) \\
&+ \sum_{i} \int_{\mathrm{d}\eta_i} q(\eta_i) \log \frac{p(\eta_i)}{q(\eta_i)} - \sum_{n=1}^{N} \sum_{z_n} q(z_n) \log q(z_n) \\
&+ \text{Extra Term} \qquad (16)
\end{aligned}
$$

where the "extra term" depends on the particular method. For the TSB prior we have,

$$
\begin{aligned}
\text{Term}_{\texttt{TSB}} = &\sum_{n=1}^{N} \sum_{z_n=1}^{T} q(z_n) \int_{\mathrm{d}\mathbf{v}} \left[\prod_{i=1}^{z_n} q(v_i) \right] \log p(z_n|\mathbf{v}) \\
&+ \sum_{i=1}^{T} \int_{\mathrm{d}v_i} q(v_i) \log \frac{p(v_i)}{q(v_i)} \qquad (17)
\end{aligned}
$$

On the other hand for the FSD prior we find,

$$
\begin{aligned}
\text{Term}_{\texttt{FSD}} = &\sum_{n} \sum_{z_n=1}^{K} \int_{\mathrm{d}\boldsymbol{\pi}} q(z_n) q(\boldsymbol{\pi}) \log p(z_n|\boldsymbol{\pi}) \\
&+ \int_{\mathrm{d}\boldsymbol{\pi}} q(\boldsymbol{\pi}) \log \frac{p(\boldsymbol{\pi})}{q(\boldsymbol{\pi})} \qquad (18)
\end{aligned}
$$

For both collapsed versions these expressions are replaced by,

$$
\text{Term}_{\texttt{CTSB/CFSD}} = \sum_{\mathbf{z}} \left[\prod_{n=1}^{N} q(z_n) \right] \log p(\mathbf{z}) \qquad (19)
$$

3.2 VB Update Equations

Given these bounds it is now not hard to derive update equations for the various methods. Due to space constraints we will refer to the papers [Blei and Jordan, 2005; Ghahramani and Beal, 2000; Penny, 2001; Yu *et al.*, 2005] for more details on the update equations for the un-collapsed methods and focus on the novel collapsed update equations.

Below we will provide the general form of the update equations where we do not assume anything about the particular form of the prior $p(\eta_i)$. The equations become particularly simple when we choose this prior in the conjugate exponential family. Explicit update equations for $q(\eta_i)$ can be found in the papers [Ghahramani and Beal, 2000; Blei and Jordan, 2005; Penny, 2001; Yu *et al.*, 2005].

For $q(\eta_i)$ we find the same update for both methods,

$$
q(\eta_i) \propto p(\eta_i) \exp \left(\sum_{n} q(z_n = i) \log p(\mathbf{x}_n|\eta_i) \right) \qquad (20)
$$

while for $q(z_n)$ we find the update

$$
\begin{aligned}
q(z_n) \propto &\exp \left(\sum_{\mathbf{z}_{\neg n}} \prod_{m \neq n} q(z_m) \log p(z_n|\mathbf{z}_{\neg n}) \right) \\
&\times \exp \left(\int_{\mathrm{d}\eta_{z_n}} q(\eta_{z_n}) \log p(\mathbf{x}_n|\eta_{z_n}) \right) \qquad (21)
\end{aligned}
$$

where the conditional $p(z_n|\mathbf{z}_{\neg n})$ is different for the FSD and TSB priors. For the TSB prior we use (10), giving the conditional

$$
p(z_n = i|\mathbf{z}_{\neg n}) = \frac{1 + N_i^{\neg n}}{1 + \alpha + N_{\geq i}^{\neg n}} \prod_{j<k} \frac{\alpha + N_{>i}^{\neg n}}{1 + \alpha + N_{\geq i}^{\neg n}} \qquad (22)
$$

where $N_i^{\neg n} = N_i - \mathbb{I}(z_n = i)$, $N_{>i}^{\neg n} = N_{>i} - \mathbb{I}(z_n > i)$ are the corresponding counts with z_n removed. In contrast, for the FSD prior we have,

$$
p(z_n = k|\mathbf{z}_{\neg n}) = \frac{N_k^{\neg n} + \frac{\alpha}{K}}{N^{\neg n} + \alpha} \qquad (23)
$$

3.3 Gaussian Approximation

The expectation required to compute the update (21) seems intractable due to the exponentially large space of all assignments for $\mathbf{z}$. It can in fact be computed in polynomial time using convolutions, however this solution still tended to be too slow to be practical. A much more efficient approximate solution is to observe that both random variables N_i and $N_{>i}$ are sums over Bernoulli variables: $N_i = \sum_n \mathbb{I}(z_n = i)$ and $N_{>i} = \sum_n \mathbb{I}(z_n > i)$. Using the central limit theorem these sums are expected to be closely approximated by Gaussian distributions with means and variances given by,

$$
\mathbb{E}[N_i] = \sum_{n=1}^{N} q(z_n = i) \qquad (24)
$$

$$
\mathbb{V}[N_i] = \sum_{n=1}^{N} q(z_n = i)(1 - q(z_n = i)) \qquad (25)
$$

$$
\mathbb{E}[N_{>i}] = \sum_{n=1}^{N} \sum_{j>i} q(z_n = j) \qquad (26)
$$

$$
\mathbb{V}[N_{>i}] = \sum_{n=1}^{N} \sum_{j>i} q(z_n = j) \sum_{k \leq i} q(z_n = k) \qquad (27)
$$

To apply this approximation to the computation of the average in (21), we use the following second order Taylor expansion,

$$
\mathbb{E}[f(m)] \approx f(\mathbb{E}(m)] + \frac{1}{2} f^{''}(\mathbb{E}[m]) \mathbb{V}[m] \qquad (28)
$$

This approximation has been observed to work extremely well in practice, even for small values of m.

3.4 Optimal Cluster Label Reordering

As discussed in section 2.1 the stick-breaking prior assumes a certain ordering of the clusters (more precisely, a size-biased ordering). Since a permutation of the cluster labels changes the probability of the data, we should choose the optimal permutation resulting in the highest probability for the data. The optimal relabelling of the clusters is given by the one that orders the cluster sizes in decreasing order (this is true since the average prior cluster sizes are also ordered). In our experiments we assess the effect of reordering by introducing algorithms O-TSB and O-CTSB which always maintain this optimal labelling of the clusters. Note that optimal ordering was not maintained in [Blei and Jordan, 2005].

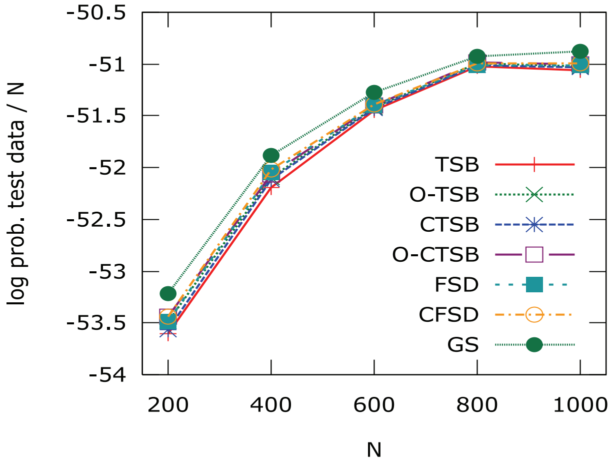

Figure 2: Average log probability per data-point for test data as a function of N.

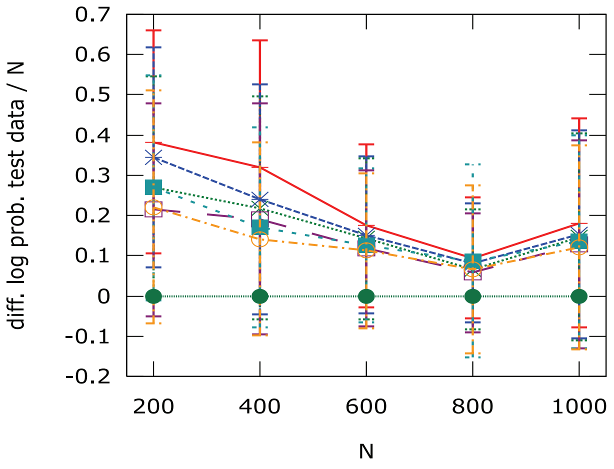

Figure 3: Relative average log probability per data-point for test data as a function of N.

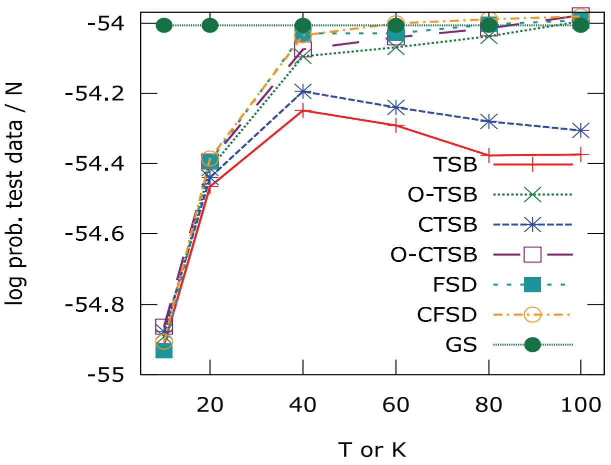

Figure 4: Average log probability per data-point for test data as a function of T (for TSB methods) or K (for FSD methods).

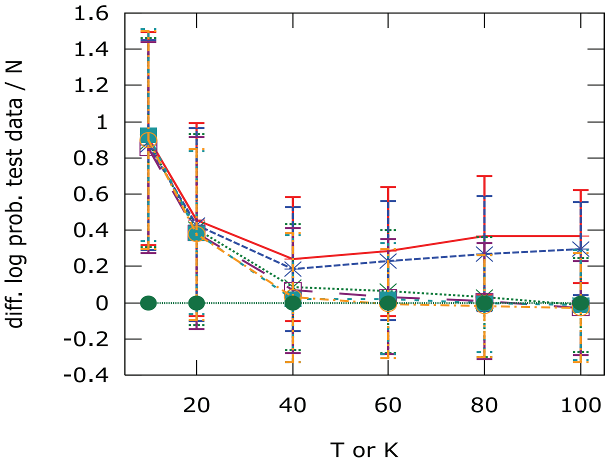

Figure 5: Relative average log probability per data-point for test data as a function of T (for TSB methods) or K (for FSD methods).

4 Experiments

In the following experiments we compared the six algorithms discussed in the main text in terms of their log-probability on held out test data. The probability for a test point, x_t, is then given by,

$$p(x_t) = \sum_{z_t} \int_{d\eta_{z_t}} p(x_t|\eta_{z_t}) q(\eta_{z_t}) \mathbb{E}[p(z_t|\mathbf{z}_{\text{train}})]_{q(\mathbf{z}_{\text{train}})}$$

where the expectation $\mathbb{E}[p(z_t|\mathbf{z}_{\text{train}})]_{q(\mathbf{z}_{\text{train}})}$ is computed using the techniques introduced in section 3.3. All experiments were conducted using Gaussian mixtures with vague priors on the parameters.

In the first experiment we generated synthetic data from a mixture of 10 Gaussians in 16 dimensions with a separation coefficient[1] $c = 2$. We studied the accuracy of each algorithm as a function of the number of data cases and the truncation level of the approximation. In figures 2 and 3 we show the results as we vary N (keeping T and K fixed at 30) while in figures 4 and 5 we plot the results as we vary T and K (keeping N fixed at 200). We plot both the absolute value of the log probability of test data and the value relative to a Gibbs sampler (GS). We 50 iterations for burn-in, and run another 200 iterations for inference. Error bars are computed on the relative values in order to subtract variance caused by the different splits (i.e. we measure variance on paired experiments).

[1] Following [Dasgupta, 1999], a Gaussian mixture is c-separated if for each pair (i, j) of components we have $||m_i - m_j||^2 \geq c^2 D \max(\lambda_i^{\max}, \lambda_j^{\max})$, where $\lambda^{\max}$ denotes the maximum eigenvalue of their covariance.

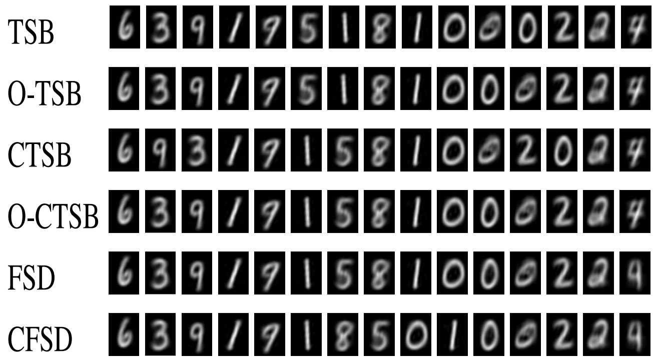

Figure 6: 15 most populated clusters found by the various algorithms in descending order of $\mathbb{E}[N_i]$. Algorithms were trained on a random subset of 10,000 images from MNIST and dimensionality reduced to 50 dimensions using PCA. Log probability of 10,000 test images are given by, $L = -574.05 \pm 0.52$ (TSB), $L = -574.03 \pm 0.53$ (O-TSB), $L = -573.90 \pm 0.54$ (CTSB), $L = -573.89 \pm 0.54$ (O-CTSB), $L = -574.06 \pm 0.50$ (FSD), and $L = -573.89 \pm 0.51$ (CFSD). Standard error over differences relative to O-CTSB are given by: $dL = -0.17 \pm 0.13$ (TSB), $dL = -0.14 \pm 0.11$ (O-TSB), $dL = -0.01 \pm 0.05$ (CTSB), $dL = -0.17 \pm 0.13$ (FSD), and $dL = -0.00 \pm 0.10$ (CFSD).

Results were averaged over 30 independently sampled training/testing datasets, where the number of test instances was always fixed at 1000.[2]

In the second experiment we have run the algorithms on subsets of MNIST. Images of size 28 × 28 were dimensionality reduced to 50 PCA dimensions as a preprocessing step. We trained all algorithms on 30 splits of the data, each split containing 5000 data-cases for training and 10,000 data-cases for testing. Truncation levels were set to 80 for all algorithms. Unfortunately, the dataset was too large to obtain results with Gibbs sampling. All algorithms typically find between 32 and 36 clusters. Results are shown in figure 4.

4.1 Discussion

In this paper we explored six different ways to perform variational Bayesian inference in DP mixture models. Besides an empirical study of these algorithms our contribution has been to introduce a new family of collapsed variational algorithms where the mixture weights are marginalized out. To make these algorithms efficient, we used the central limit theorem to approximate the required averages.

We can draw three conclusions from our study. Firstly, there is very little difference between variational Bayesian inference in the reordered stick-breaking representation and the finite mixture model with symmetric Dirichlet priors. Secondly, label reordering is important for the stick-breaking representation. Thirdly, variational approximations are much more efficient computationally than Gibbs sampling, with almost no loss in accuracy.

We are currently working towards models where the parameters $\boldsymbol{\eta}$ are marginalized out as well. We expect this to have a more significant impact on test accuracy than the current setup which only marginalizes over $\boldsymbol{\pi}$, especially when clusters are overlapping. Unfortunately, it seems this will come at the cost of increased computation.

Collapsed variational inference has also been applied to LDA models [Teh *et al.*, 2006], where preliminary results indicate significant performance improvement. We are currently also exploring collapsed variational inference for hierarchical DP models [Teh *et al.*, 2004].

References

[Attias, 2000] H. Attias. A variational bayesian framework for graphical models. In *NIPS*, volume 12, 2000.

[Blei and Jordan, 2005] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, 1(1):121–144, 2005.

[Dasgupta, 1999] S. Dasgupta. Learning mixtures of gaussians. In *Fortieth Annual IEEE Symposium on Foundations of Computer Science*, 1999.

[Ghahramani and Beal, 2000] Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. In *NIPS*, volume 12, 2000.

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

[Ishwaran and Zarepour, 2002] H. Ishwaran and M. Zarepour. Exact and approximate sum-representations for the Dirichlet process. *Can. J. Statist.*, 30:269–283, 2002.

[Kurihara *et al.*, 2006] K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational dirichlet process mixtures. In *NIPS*, volume 19, 2006.

[Moore, 1998] A. Moore. Very fast EM-based mixture model clustering using multiresolution kd-trees. In *NIPS*, volume 10, 1998.

[Penny, 2001] W.D. Penny. Variational bayes for d-dimensional gaussian mixture models. Technical report, Department of Cognitive Neurology, University College London, 2001.

[Porteous *et al.*, 2006] I. Porteous, A. Ihler, P. Smyth, and M. Welling. Gibbs sampling for (coupled) infinite mixture models in the stick-breaking representation. In *UAI*, Cambridge, MA, 2006.

[Teh *et al.*, 2004] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. In *NIPS*, volume 17, 2004.

[Teh *et al.*, 2006] Y.W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *NIPS*, volume 19, 2006.

[Verbeek *et al.*, 2003] J. Verbeek, J. Nunnink, and N. Vlassis. Accelerated variants of the em algorithm for gaussian mixtures. Technical report, University of Amsterdam, 2003.

[Yu *et al.*, 2005] K. Yu, S. Yu, and V. Tresp. Dirichlet enhanced latent semantic analysis. In *Conference in Artificial Intelligence and Statistics*, 2005.

[2] For N=200 and T or K=40, TSB, O-TSB, CTSB, O-CTSB, FSD, CFSD and GS took 0.48, 0.70, 0.84, 1.22, 0.67, 1.16 and 1,019 seconds on average, respectively. Note that the computational complexities of the variational algorithms are the same.