Beam Search based MAP Estimates for the Indian Buffet Process

Piyush Rai PIYUSH@CS.UTAH.EDU
School of Computing, University of Utah, Salt Lake City, UT, USA

Hal Daumé III HAL@UMIACS.UMD.EDU
Department of Computer Science, University of Maryland, College Park, MD , USA

Abstract

Nonparametric latent feature models offer a flexible way to discover the latent features underlying the data, without having to *a priori* specify their number. The Indian Buffet Process (IBP) is a popular example of such a model. Inference in IBP based models, however, remains a challenge. Sampling techniques such as MCMC can be computationally expensive and can take a long time to converge to the stationary distribution. Variational techniques, although faster than sampling, can be difficult to design, and can still remain slow on large data. In many problems, however, we only seek a maximum a posteriori (MAP) estimate of the latent feature assignment matrix. For such cases, we show that techniques such as beam search can give fast, approximate MAP estimates in the IBP based models. If samples from the posterior are desired, these MAP estimates can also serve as sensible initializers for MCMC based algorithms. Experimental results on a variety of datasets suggest that our algorithms can be a computationally viable alternative to Gibbs sampling, the particle filter, and variational inference based approaches for the IBP, and also perform better than other heuristics such as greedy search.

1. Introduction

Automatically discovering the latent feature representation of data is an important problem in various data analysis tasks. The latent feature assignments for a given set of observations can be expressed by a (potentially sparse) binary matrix. Such representations achieve two-fold benefits: (1) understanding hidden or causal structures underlying the data (Wood et al., 2006; Meeds et al., 2006; Rai & Daumé III, 2008) by giving it a parsimonious representation, and (2) using these latent representations in prediction settings (e.g., (Rai & Daumé III, 2008)). Often in such settings, the number of latent features is not known *a priori*. Nonparametric Bayesian methods (Orbanz & Teh, 2010) offer an elegant solution to this issue by defining a model having an unbounded complexity and allowing the data to figure out the right complexity by itself.

The Indian Buffet Process (IBP) (Ghahramani et al., 2007) is one such nonparametric prior distribution over infinite, sparse binary matrices. The IBP allows discovering the set of latent features possessed by each observation, without having to specify the number of latent features K in advance. Unfortunately, the combinatorially complex nature of the IBP (search over all possible binary feature assignment matrices) poses significant challenges during inference in the IBP based models. MCMC based approaches such as Gibbs sampling (Ghahramani et al., 2007) are traditionally used in these models, which tend to be computationally expensive and may take long to converge. Another alternative is to use variational methods (Doshi-Velez et al., 2009b). Although faster than the sampling based methods, these can be difficult to design and implement, and can potentially run into local optima issues.

Sampling based methods such as MCMC produce samples from the posterior distribution. However, in many applications we only require the *maximum a posteriori* (MAP) sample, discarding all other samples. This naturally leads to the following question: *If all we care about is a single MAP assignment, why not find one directly?* Furthermore, note that although sampling and variational methods *aim* to explore the full posterior over the latent feature matrix, they may not be

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

well-suited for searching a posterior mode: Sampling may take too long to mix and get close to the maxima; variational methods may not be able to find the true maxima due to their inherent local maxima problem. In this paper, we propose search algorithms such as A^* and beam search (Russell & Norvig, 2003) for finding *approximate* MAP estimate of the latent feature assignment matrix. Our approach can be a viable and more efficient alternative to sampling or variational approaches if only the MAP estimate is required. If samples from the true posterior are desired then the search based MAP estimate can serve as a sensible initializer for MCMC, resulting in faster convergence.

2. Infinite Latent Feature Model

Given an $N \times D$ matrix X of N observations having D dimensions each, the latent feature model represents X as $ZA+E$. Here Z is an $N \times K$ binary matrix (with $K \ll D$) denoting which latent features are present in each observation, A is a $K \times D$ matrix consisting of feature scores, and E consists of observation specific noise. A crucial issue in these models is the choice of K, the number of latent features. The Indian Buffet Process (IBP) (Ghahramani et al., 2007) defines a prior distribution on the binary matrix Z such that it can have a potentially unbounded (i.e., infinite) number of columns, and offers a principled way to select K automatically from the data.

The IBP has a nice culinary analogy of N customers coming to an Indian buffet and making selections from an infinite array of dishes. In this analogy, customers represent observations (rows of X and Z) and dishes represent latent features (columns of Z). Customer 1 selects $Poisson(\alpha)$ dishes to begin with, where α is an IBP hyperparameter. Thereafter, each incoming customer n selects an existing dish k with a probability m_k/n, where m_k denotes how many previous customers chose that particular dish. The customer n then goes on further to additionally select $Poisson(\alpha/n)$ new dishes. This process generates a binary matrix Z with rows representing customers and columns representing dishes. The IBP further has the exchangeability property that the order in which the customers enter the buffet does not affect the distribution of Z. The IBP defines the following probability distribution over the *left-ordered-form* of Z (invariant to latent feature ordering; see (Ghahramani et al., 2007) for details):

$$P([Z]) = \frac{\alpha^K}{\prod_{h=1}^{2^N-1} K_h!} e^{(-\alpha H_N)} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!}$$

where H_N is the N^{th} harmonic number, K_h is the number of columns in Z with binary representation h, and $m_k = \sum_i Z_{ik}$. K is the number of non-zero columns in Z.

```
function IBPSearch
input: a scoring function $g$, beam size $b$, data $X_{1:N}$
output: IBP matrix $Z$
initialize max-queue: $Q \leftarrow [\langle\rangle]$
while $Q$ is not empty do
  remove the best scoring candidate $Z$ from $Q$
  if $|Z| = N$ then return $Z$
  for all possible assignments $Z_{N^0}$ for the next (say $N^0$-th) customer (i.e., each of the $2^K$ possibilities from existing dishes, and for each possibility 0 and $\max\{1, \lceil\alpha/N^0\rceil - 1\}$ new dishes) do
    let $Z^0 = [Z; Z_{N^0}]$
    compute the score $s = g(Z^0, X)$
    update queue: $Q \leftarrow$ Enqueue($Q$, $Z^0$, $s$)
  end for
  if $b < \infty$ and $|Q| > b$ then
    Shrink queue: $Q \leftarrow Q_{1:b}$
        (drop lowest-scoring elements)
  end if
end while
```

Figure 1. The generic IBP search algorithm (takes the scoring function as input).

In this paper, we consider models of the form $X = ZA + E$ (e.g., the linear-Gaussian model (Ghahramani et al., 2007)) where A can be integrated out and thus $P(X|Z) = \int P(X|Z,A)P(A)dA$ can be represented in closed form, or can be approximated efficiently. Here, we do not describe computing A but, given Z, it is easy to compute in these models.

3. Search based MAP Estimate for IBP

Our beam-search algorithm (Figure 1) for the IBP takes as input the set of observations, a scoring function g, and a maximum beam size b. The algorithm maintains a max-queue of candidate latent feature assignment matrices. Each of these matrices on the queue is associated with a score on the basis of how likely it is to maximize the posterior probability of the *complete* Z given X. This essentially means how likely it is to being the eventual MAP estimate once we have seen all the observations. The maximum beam size specifies the maximum number of candidates allowed on the queue at any time. At each iteration, the highest scoring candidate Z is removed from the queue, and is expanded with the set of all possible feature assignments for the next (say N^0-th) observation. For the possible expansions, we consider 2^K possibilities for assigning the existing dishes and, for each such possibility, 0 and $\max\{1, \lceil\alpha/N^0\rceil - 1\}$ new dishes (note: $\lceil\alpha/N^0\rceil - 1$ is the *mode* of the number of new dishes chosen by the N^0-th customer in the IBP culinary analogy). Our algorithm therefore explores matrices Z of sizes *up to* $N \times \sum_{n=1}^{N} \max\{1, \lceil\alpha/N^0\rceil - 1\}$, but this is a reasonable approximation since the number of latent features is typically much smaller than N or D. Scores are computed for each of the new candidates and these candidates are placed in the queue. If the beam size is not infinite then we also drop the lowest scoring elements so as to maintain the maximum queue size. We stop at the point when the number of rows in the matrix removed from the queue equals the total number of observations.

Scoring of the candidate latent feature assignment matrices constitutes an important aspect of our search algorithms. Recall that finding the MAP estimate requires finding Z that maximizes the posterior probability of Z given X, $P(Z|X)$, which is proportional to the joint probability $P(Z, X)$. However, since our algorithm processes one observation at a time (in an online fashion), at any point having seen N^0 observations, we can only have an upper bound on the joint probability of all N observations. Since the joint probability $P(Z, X)$ can be again factored as $P(Z)P(X|Z)$, an upper bound on $P(Z, X)$ can thus be obtained by independently upper-bounding the prior probability:

$$P(Z) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k - 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

where $m_k = \sum_i Z_{ik}$, and the likelihood $P(X|Z)$, both given the first N^0 observations. In fact, as we shall show (Section 4), it is possible to even explicitly upper bound the prior term. Unfortunately, the same is not true for the likelihood term (as it also involves the future observations and their latent feature assignments), and we therefore propose several heuristics for upper bounding the likelihood term (Section 5). The sum (assuming probabilities are expressed on log scale) of these two terms is the scoring function.

The search algorithm is guaranteed to find the optimal MAP feature assignment matrix if the beam size is infinite and the scoring function g is *admissible*. Being admissible means that it should *over-estimate* the posterior probability of *best possible* feature assignment Z that agrees with Z^0 on the first N^0 observations. Denoting the condition as $Z|N^0 = Z^0$ as the *restriction* of Z to the first N^0 elements, admissibility can be written formally as:

$$g(Z^0, X) \geq \max_{Z:Z|N^0=Z^0} P(Z, X)$$

Although the admissible scoring functions provably lead to optimal MAP estimates, the NP-hardness of the MAP problem implies that these can be inefficient (in terms of enqueue/dequeue operations on the queue; a large gap between these two numbers would mean that it takes too long to search for the optimal candidate). For efficiency reasons, it is often useful to have scoring functions that occasionally *under-estimate* the true posterior probability, and are therefore *inadmissible*. In fact, as described in Section 5, our proposed scoring functions are not guaranteed to be admissible in general, but they lead to efficient approximate MAP estimates for the Z matrix (see the experimental section for evidence supporting this).

Our search algorithm is akin to the A^* search (Russell & Norvig, 2003) where we optimize a *path-cost-so-far* function plus a *cost-to-goal* function. In our case, we rank a candidate feature assignment matrix by computing its score that is a summation of the joint probability $P(X, Z)$ up to first N^0 observations (similar to the path-cost-so-far), and an *upper bound* on the joint probability corresponding to the remaining observations (similar to the cost-to-goal). Since the joint probability can be factored into the prior and the likelihood terms, we next show in Section 4 and Section 5 how each of these can be upper bounded. In keeping with the culinary metaphor of IBP, in the rest of the exposition, we will occasionally refer to observations as customers, and features as dishes.

4. Upper Bounding the Prior

Given the customer-dish assignment Z^0 for the first N^0 customers, it is possible to explicitly compute the dish assignments for the remaining customers that maximizes the probability $P(Z)$. For this maximization, we need to consider two cases for the remaining customers: (a) maximization w.r.t. the already selected dishes, and (b) maximization w.r.t. the new dishes.

Upper bounding w.r.t. already selected dishes: Given an $N^0 \times K$ matrix Z^0 for the first N^0 customers, if one were to maximize the IBP prior $P(Z)$, then the $(N^0 + 1)^{th}$ customer would choose an already selected dish k only if it was chosen previously by more than half the customers (i.e., the *majority*). Let us denote this event by a random variable $x_k = \mathbb{I}_{(m_k > N^0/2)}$, where $\mathbb{I}$ is the indicator function and m_k is the number of previous customers who chose the k^{th} dish. Now, to maximize $P(Z)$, all subsequent customers would also make the same choice as the $(N^0+1)^{th}$ customer (since the customers making that choice will continue to remain in the majority). To derive the *probability* of this event happening, we appeal to the exchangeability of the IBP and can assume that the $(N^0+1)^{th}$ customer comes at the end after the remaining $(N-N^0-1)$ customers (who either all select or all skip the dish k). Therefore the probability that the $(N^0+1)^{th}$ customer *selects* dish k is $p_k = (m_k + (N-N^0-1))/N$, and the probability that this dish is skipped $1-p_k$. Since all the $(N-N^0)$ customers make the identical choice in selecting/skipping this dish, the random variable $x_k \in \{0,1\}$ and p_k take on the same values for each customer. This leads to a score w.r.t. dish k:

$$s_k = [p_k^{x_k}(1-p_k)^{(1-x_k)}]^{(N-N^0)}$$

which is a product of $(N-N^0)$ binomials. The total score for the maximization w.r.t. the existing dishes is given by the *product* (or the log sum if using log probabilities) of individual scores for each of the existing dishes.

Upper bounding w.r.t. the new dishes: In the IBP culinary metaphor, the n^{th} customer selects $Poisson(\alpha/n)$ number of new dishes so the prior would be maximized if customer n selects a number of dishes equal to the *mode* of this number which is $\lfloor\alpha/n\rfloor$. The score contribution of this part for $P(Z)$ is given by:

$$\prod_{n=N^0+1:N} \frac{(\alpha/n)^{\lfloor\alpha/n\rfloor!}\exp(-\alpha/n)}{\lfloor\alpha/n\rfloor!}$$

The part of the above product involving the exp terms just requires computing a harmonic mean of $(N-N^0)$ numbers. For the terms involving $\lfloor\alpha/n\rfloor$, we only need to care about those for which $\lfloor\alpha/n\rfloor > 0$. This computation is inexpensive since α is usually small and therefore $\lfloor\alpha/n\rfloor$ quickly goes to zero .

5. Upper Bounding the Likelihood

Unlike the prior term, an explicit maximization is not possible for the likelihood because the future observations would not have been assigned any latent features yet, precluding the associated likelihood computation. We propose here several heuristics for approximating the likelihood of future observations.

5.1. A Trivial Function

Given the matrix Z^0 having N^0 many rows, a possible trivial upper bound on $P(X|Z)$ can be obtained by only considering the likelihood over the first N^0 observations. This function is given by:

$$g_{Trivial}(X \mid Z^0) = P(X_{1:N^0} \mid Z^0)$$

For discrete likelihood distributions (e.g., multinomial likelihood), the *true* likelihood of each future observation is upper bounded by 1. Therefore the above function would be a trivial upper bound on $P(X|Z)$, since it assigns a probability one to the likelihood term of each future observation. With an infinite beam size, this admissible function is guaranteed to find the optimal MAP estimate. Note that this would however not be true for continuous likelihood distributions, e.g., Gaussian likelihood which is actually a density (not a probability) upper bounded by $(2\pi\sigma_X^2)^{-1/2}$. Unless the data variance σ_X is such that $(2\pi\sigma_X^2)^{-1/2} \leq 1$, admissibility is not guaranteed in such cases, and the search would not be guaranteed to find the global optimal solution. Moreover, as discussed earlier in Section 3, even though the trivial function is admissible in certain cases and may find the optimal solution, the bound tends to be quite loose which can make the search inefficient (see empirical evidence in the experiments section).

5.2. An Inadmissible Function

Another possibility is to use a function which is significantly tighter (i.e., better approximation to the true likelihood), but not admissible in any of the cases. Therefore the search is no longer guaranteed to find the global optimal solution. However, since it is tighter, it is much more efficient to run, and can find approximate solutions much more quickly. This *inadmissible* function is given by:

$$g_{Inad}(X \mid Z^0) = P(X \mid [Z^0; Z_{N^0+1:N}])$$

where $Z_{N^0+1:N}$ is a matrix of size $(N-N^0)\times(K+N-N^0)$ such that each future customer $n \in [N^0+1,\ldots,N]$ gets assigned a single, its own new dish. Here $[Z^0; Z_{N^0+1:N}]$ denotes row-wise concatenation with appropriate padding of Z^0 and $Z_{N^0+1:N}$ with zeros. This is an inadmissible heuristic since it is always preferable to instead assign the same set of dishes to two customers if both are identical, a fact which this function does not take into account.

5.3. A Clustering Based Function

Even though the trivial function discussed above is admissible in certain cases (i.e., discrete likelihood distributions), the upper bound is very loose since it does not take into account the feature assignments of any of the future observations, and the search would therefore be inefficient. The inadmissible function, on the other hand, assigns a single new dish to each future customer which may not mirror the likelihood of future observations that closely. Our next proposal aims to find a middle ground by trying to account for the probable dish selection by the remaining customers.

One way to incorporate the dish assignment of future customers in the likelihood term is to first do a *coarse level* of feature assignment. Given the set of observations $X = [X_1, \ldots, X_N]$, we first run a clustering algorithm with a small number of clusters. Having obtained a clustered representation of the data, we pick one representative point from each cluster and run the IBP search algorithm (using the trivial scoring function described above) on these cluster representative observations. This gives us a *coarse* feature assignment for the representative points. We then run the IBP search on the full data and, while computing the likelihood (heuristic) of a future observation n, we use the same set of latent features for this observation as assigned to the representative data point of the cluster it belongs to.

A number of other alternatives for likelihood maximization, though not evaluated here, can be tried as well. For example, for computing the upper bound on the likelihood of a future observation, we can assign it the same set of latent features as its nearest neighbor observation from among the observations seen thus far. Alternatively, one can do an OR of the latent features of the K-nearest neighbors of the future observation, and use the resulting bit vector as the set of feature assignments for this observation.

6. Experiments

We report experimental results on a variety of datasets (both synthetic and real), and compare the search based approaches against a number of baselines. Our results are on two type of tasks: (1) latent factor analysis (Rai & Daumé III, 2008), and (2) factor regression (West, 2003; Rai & Daumé III, 2008) which uses the factors for making predictions in classification or regression settings (we experiment with classification setting). For the factor analysis task, we report the joint log probability scores and the time taken, and for the factor regression task, we report the predictive accuracies on a held-out test data.

6.1. Baselines and experimental setup

The baselines we compare against are uncollapsed Gibbs sampling (Ghahramani et al., 2007), infinite variational inference (Doshi-Velez et al., 2009b), and particle filtering (Wood & Griffiths, 2007) for the IBP. In addition, we also briefly discuss a comparison with a greedy search based approach (Section 6.7). The variational inference was given 5 random restarts to avoid the issue of local optima (the reported time is the average time taken for a *single run*). The particle filter was run with a varying number of particles (500-5000) and the reported results are the best achieved with minimum possible number of particles. We would like to note here that we also compared with the semi-collapsed Gibbs sampler for IBP (Doshi-Velez & Ghahramani, 2009) but the results and the running times were very similar to the uncollapsed Gibbs so we included only the uncollapsed version in our experiments. The uncollapsed version has the same time complexity as the semi-collapsed version (linear in the number of observations). Although the uncollapsed version is sometimes known to mix slowly, we did not observe this in our experiments. For our search based approaches, we used small beam sizes (10-20) which seemed to be enough for our experiments.

6.2. Block-images dataset

In our first experiment, we applied our search based approach to the block-image dataset with known ground truth, generated in a manner akin to (Ghahramani et al., 2007) using a linear-Gaussian model of the data: $X = ZA + E$. The feature score matrix A has a zero mean Gaussian prior: $A \sim \mathcal{N}or(0, \sigma_A^2)$, and the noise as well is Gaussian: $E \sim \mathcal{N}or(0, \sigma_X^2)$. Our dataset consists of twenty 4 × 4 synthetic block-images generated by combining four different 4 × 4 latent images. The latent feature assignment matrix Z is 20 × 4. More importantly, we note that Z was *not* generated from an IBP prior. Each generated image had Gaussian noise with $\sigma_X = 0.1$ added to it. We then ran our search based approaches and various baseline approaches on this data. The trivial, cluster-based, and the inadmissible approaches finish reasonably fast, taking a time of 1.02 seconds, 0.86 seconds, and 0.45 seconds respectively, suggesting that the inadmissible search is the fastest among all (the number of enqueued/dequeued elements, though not reported to conserve space, were also the smallest for this method). In comparison, Gibbs sampling took 3.30 seconds, particle filter 0.98 seconds, and the infinite variational inference (Doshi-Velez et al., 2009b) took 3.73 seconds to finish (truncation level was set to 12). All approaches recovered the ground truth latent features on this data.

6.3. E-Coli data

The E-Coli dataset is a gene-expression dataset with known gene-pathway loadings which is a sparse 50 × 8 binary matrix ($K = 8$) (Rai & Daumé III, 2008). This is a semi-real dataset; the gene-factor connectiv-

Table 1. Results on the E-coli data

	K	Time (sec)	logP(X,Z)
Gibbs Sampling	6	49.8	**-4681**
Particle Filter	7	17.8	-5369
Infinite Variational	3	12.1	-6875
Trivial	**8**	72.5	-5887
Cluster Based	**8**	15.5	-5759
Inadmissible	**8**	**10.3**	-5865

Table 2. Latent factor based classification results

	Sonar		**Scene**	
	Acc	K	Acc	K
Gibbs	70.9 (±4.8)	6	77.6 (±0.9)	6
Particle Filter	52.4 (±4.2)	6	77.8 (±1.3)	10
Infinite Variational	68.5 (±5.6)	10	74.3 (±2.1)	9
Trivial	72.4 (±3.9)	7	76.2 (±1.7)	7
Cluster Based	71.5 (±3.6)	7	77.8 (±2.1)	6
Inadmissible	67.1 (±4.9)	5	76.9 (±3.2)	6

ity network (binary Z matrix) is taken from a real dataset and the observations are simulated using this network using a linear-Gaussian model. We generated 50 observations with 100 dimensions each. The number of latent features, time taken, and log-joint probabilities reported by our search based approaches and the other baselines are given in Table 1. As we see, our search based approaches successfully recover the correct number of latent features (8) in the data, and are reasonably faster (with the inadmissible approach being the fastest) than the other baselines. The variational inference, although comparable to search in terms of speed, severely underestimates the number of latent features, possibly due to getting trapped in a local optima. In our experiment, we set the beam size to 10 in all the search based approaches. The IBP parameter α was set to 3 and the hyperparameters (the noise variance σ_X and latent feature variance σ_A) were set based on the data variance, for all the algorithms, akin to the way in (Doshi-Velez et al., 2009b; Doshi-Velez & Ghahramani, 2009).

6.4. Scalability

Next, we demonstrate the scalability of the search based algorithms with the number of observations. We report experiments on one synthetic and one real-world dataset. The synthetic dataset was generated using the IBP Prior with $\alpha = 1$ and linear Gaussian model of the data with noise variance $\sigma_X = 0.1$. The generated dataset consists of 1000 data points, each with 100 dimensions, and the number of latent features K is 4. We varied the number of observations from 200 to 1000 with increments of 200. For the real-world dataset, we take the 50×100 E-coli data and vary the number of observations from 10 to 50. The timings and log-joint probabilities for the synthetic and E-coli datasets are shown in Figure 2. As the figures show, the search based approaches are the fastest on both the datasets (except for the trivial heuristic on E-Coli data). On the synthetic data, all the search approaches actually recover the ground truth (the log-joint probabilities of all search based approaches therefore look the same). Also, although the timings are roughly the same for all search based approaches, the inadmissible search did the fewest number of enqueue/dequeue operations, and was therefore the fastest. Among the other baselines, the variational inference is the fastest one but it fails to recover good solutions most of the time (as measured by the log-joint probability, and also the number of latent features discovered). The particle filter, although scaled well on small data regimes (E-Coli data), scaled poorly for large datasets, as can be seen by its (lack of) scalability on the synthetic data.

6.5. Factor Regression

Next, we apply the various methods on real-world binary classification datasets to extract latent factors and use them to train a classification model (akin to (West, 2003; Rai & Daumé III, 2008)). We use two real-world datasets for the classification tasks: the aspect-angle dependent sonar signals dataset and the scene classification dataset from the UCI Machine Learning Repository. The sonar signal dataset consists of 208 examples having 60 features each. The scene classification dataset is actually a multi-label dataset with 2407 examples having 294 features each; we chose the 7^{th} label as a prediction task. Since the feature assignment matrix is binary and the latent factors we care about are real-valued, we applied all the algorithms on the transposed $D \times N$ data matrix. The matrix Z is $D \times K$ in this case, and we treat the $K \times N$ real-valued, feature score matrix A as the factor matrix (N examples with K real-valued features each) used to train the classification model. For the search based algorithms, we compute A by drawing a sample from its posterior given Z. After the feature extraction stage, we split the data into two equal parts (training and test), train an SVM classifier (with linear kernel) and then apply the learned classifier on the test data. We experiment with 200 random splits of training and test data and report the average and standard deviation of the accuracies achieved by various methods. As the results in Table 2 show, the search based approaches achieve prediction performance that, in most cases, is competitive (or better) than Gibbs sampling. At the same time, search finished much faster than sampling in the latent factor analysis step of the task.

6.6. (Approximate) MAP as an initializer

The search based approach yields a MAP estimate. In many cases, however, we care about the full posterior.

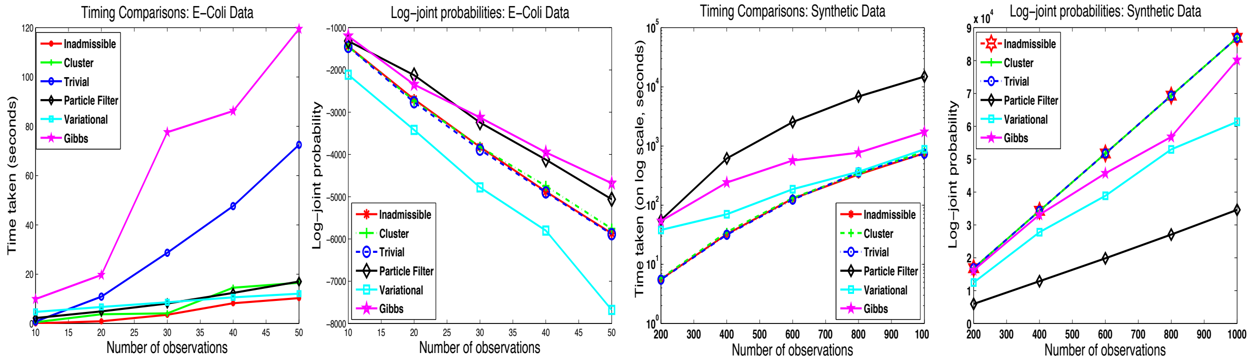

Figure 2. Scalability results of various algorithms for E-Coli and Synthetic datasets

In such cases, the approximate MAP estimate found by our search based algorithms can serve as a sensible initializer to the sampling based approaches. As an illustration, we ran an uncollapsed Gibbs sampler by using random initialization and the search based MAP initialization, and monitored the joint likelihood over time. As we see in Fig 3, the MAP initialized Gibbs sampler localizes itself in the high probability region quite early on, as compared to randomly initialized sampler which takes much longer to attain similar values of the joint likelihood. The extra overhead of doing the search to get the MAP estimate is much smaller than the overall time taken by the Gibbs sampler.

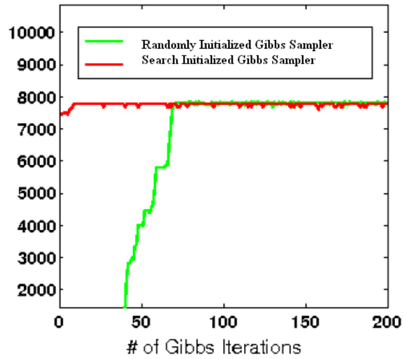

Figure 3. Log-likelihood scores for random vs search based MAP initialized Gibbs Sampler

6.7. Comparison with Greedy Search

We also compared our beam search based approach with a greedy search heuristic which works by selecting, for the $(N^0+1)^{th}$ observation, the feature assignment Z_{N^0+1} that maximizes the posterior probability up to this observation, i.e., $P([Z^0; Z_{N^0+1}] \mid X_{1:N^0+1})$. Note that this heuristic is similar to the one proposed in (Wang & Dunson, 2011) for the Dirichlet Process Mixture Model. Also, the greedy search approach is akin to beam search with the trivial heuristic, but without the explicit prior term maximization as we do in Section 4 (it only considers the prior $P([Z^0; Z_{N^0+1}])$ up to the N^0+1 observations) and a beam size of 1. Due to space limit, we do not report the full experimental results here but we found that, on the block-images dataset (Section 6.2), greedy search ran much slower than our inadmissible approach, ran almost as fast as the trivial heuristic, but inferred a much larger value of K than the ground truth (and lower log-likelihood scores). Moreover, the greedy search that only considers the posterior probability up to the current observation (ignoring the future observations) is not expected to do well if the number of observations is very large.

7. Related Work

In this section, we review previous work on inference in IBP based models, some of which were used as baselines in our experiments. One of the first attempts to scale inference in IBP based models to large datasets was the particle filter (Wood & Griffiths, 2007) for IBP. Particle filters are somewhat similar in spirit to our approach since a particle filter can be considered as doing a stochastic beam search. The particle filter can process one observation at a time. However, the particle filter samples each row of Z from the prior and the naïve sequential importance resampling scheme does not perform very well on datasets having a large number of observations (which is perhaps the reason behind the poor performance of particle filter in our experiments). Besides, particle filters are known to suffer from the sample impoverishment problem and need to make multiple passes over the data to deal with this issue. Among the sampling based approaches, (Doshi-Velez & Ghahramani, 2009) proposed a fast collapsed Gibbs sampler to address the slow mixing issue of the uncollapsed Gibbs sampler. Other sampling based approaches include the Metropolis split-merge proposals (Meeds et al., 2006), and slice sampling (Teh et al., 2007). Parallelization of the sampling based inference for the IBP has also been attempted (Doshi-Velez et al., 2009a).

Deterministic variational inference can be an efficient alternative to sampling in IBP based models. One such approach was proposed in (Doshi-Velez et al., 2009b) who proposed a variational inference algorithm for IBP which is based on the truncated stick-breaking approximation. Our search based approach for inference is also deterministic and is similar in spirit to (Daumé III, 2007) who applied beam search algorithms for finding MAP estimates in Dirichlet Process mixture models. However, we note that the combinatorial problem posed by the IBP is even more challenging than the DP since the former looks at the space of $\mathcal{O}(2^{NK})$ possible feature assignments as opposed to the latter where this space is $\mathcal{O}(K^N)$ possible clusterings of the data.

8. Discussion and Conclusion

In this paper, we have presented a general, search-based framework for MAP estimates in the nonparametric latent feature models. There are several aspects of the proposed algorithm that can be improved even further. Note that when a candidate is removed from the queue and expanded with the possible feature assignments for the next observation, we need to consider all 2^K possible candidates, compute their scores, and place them on the queue. This can be expensive for cases where K is expected to be large. An alternative to this would be to modify the proposed beam search by expanding along the *columns* of the Z matrix for a given row, considering one dish at a time (this would amount to a *search-within-search* procedure). Such a modification is expected to make search even faster. Besides, the heuristics used for likelihood maximization are critical to getting tighter bounds for the posterior and it would be interesting to consider other possible heuristics that result in even tighter even bounds. Another possibility is to estimate the hyperparameters (IBP hyperparameter α and the variance hyperparameters σ_X and σ_A which are currently set of a fixed value); for examples, as is done in (Wang & Dunson, 2011). Finally, although in the paper we showed the conjugate case as an example (where we do not care about A), conjugacy is not necessary for our approach to be applicable. If the A matrix can't be integrated out due to the non-conjugate prior, we can explicitly represent it at each step of the search algorithm by also computing the MAP assignment for A, given Z (for example, by running a few steps of some gradient based optimizer), or by running a few Metropolis-Hastings steps for A, given Z.

Acknowledgement: We thank the anonymous reviewers for their feedback. This work was partially funded by NSF grant IIS 0712764.

References

Daumé III, H. Fast Search for Dirichlet Process Mixture Models. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

Doshi-Velez, F. and Ghahramani, Z. Accelerated Sampling for the Indian Buffet Process. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

Doshi-Velez, F., Knowles, D., Mohamed, S., and Ghahramani, Z. Large Scale Nonparametric Bayesian Inference: Data Parallelisation in the Indian Buffet Process. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2009a.

Doshi-Velez, F., Miller, K. T., Gael, J. V., and Teh, Y. W. Variational Inference for the Indian Buffet Process. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009b.

Ghahramani, Z., Griffiths, T. L., and Sollich, P. Bayesian Nonparametric Latent Feature Models. In *Bayesian Statistics 8*, pp. 201–226, 2007.

Meeds, E., Ghahramani, Z., Neal, R., and Roweis, S. Modeling Dyadic Data with Binary Latent Factors. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2006.

Orbanz, P. and Teh, Y. W. Bayesian Nonparametric Models. In *Encyclopedia of Machine Learning*. Springer, 2010.

Rai, P. and Daumé III, Hal. The Infinite Hierarchical Factor Regression Model. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2008.

Russell, Stuart J. and Norvig, Peter. *Artificial Intelligence: a modern approach*. Prentice Hall, 2nd international edition edition, 2003.

Teh, Y. W., Görür, D., and Ghahramani, Z. Stick-breaking Construction for the Indian Buffet Process. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

Wang, Lianming and Dunson, D. B. Fast Bayesian Inference in Dirichlet Process Mixture Models. In *Journal of Computational and Graphical Statistics 20(1)*, pp. 196–216, 2011.

West, M. Bayesian Factor Regression Models in the "Large p, Small n" Paradigm. In *Bayesian Statistics 7*, pp. 723–732, 2003.

Wood, F. and Griffiths, T. L. Particle Filtering for Nonparametric Bayesian Matrix Factorization. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2007.

Wood, F., Griffiths, T. L., and Ghahramani, Z. A Non-parametric Bayesian Method for Inferring Hidden Causes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.