Exchangeable Inconsistent Priors for Bayesian Posterior Inference

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

Ian Porteous
Google Inc.
Kirkland, WA, USA
Email: ian.porteous@gmail.com

Kenichi Kurihara
Google Inc.
Tokyo, Japan
Email: kenichi.kurihara@gmail.com

***Abstract*—Nonparametric Bayesian methods offer a convenient paradigm to deal with uncertain model structure. However, priors such as the (hierarchical) Dirichlet process prior on partitions and the Indian buffet process prior on binary matrices are not always flexible enough to express our prior beliefs. We propose a much larger family of nonparametric exchangeable priors by relaxing the concept of consistency. We discuss the consequences of this point of view and propose novel ways to specify and learn these priors. In particular, we introduce new flexible priors and inference procedures to extend the DP, HDP and IBP models. An experiment on text data illustrates how flexible priors can be useful to increase our modeling capabilities.**

I. Introduction

The Dirichlet process (DP) has proven to be a very useful model for applications related to clustering and density estimation. One can think of it as a prior over all possible partitions of the data. It's popularity can at least be partially explained by the fact that it offers an elegant way to estimate the number of clusters in the data by means of an efficient inference procedure (e.g. collapsed Gibbs sampling with split-merge moves [1], [2]). However, does the DP really always reflect our prior beliefs about how the data is partitioned into groups? In [3] the authors argue that the DP tends to create many very small clusters, an effect that persists in the posterior.

Why is it then that we seem to be married to using DPs and Pitman-Yor processes (PYP)? To answer that we recall that the DP, and more generally the PYP, satisfy the following two conditions [4], [3]:

1) *Exchangeability*: The probability distribution is invariant under permuting the labels of the data-items.
2) *Consistency (Heritability)*: The probability distribution for n data-items is the same as the distribution for $N > n$ data items and subsequently marginalizing down to n data-items, $\forall n, N \in \mathbb{N}$.

According to Kolmogorov's extension theorem, these conditions are sufficient to guarantee the existence of a stochastic process. Unfortunately, it is not easy to find analytic expressions that satisfy these two properties and moreover, it is unclear whether they are necessary ingredients for viable machine learning methods. Take for instance the problem of clustering data-items into groups. The DP is a popular tool in this domain because it defines a prior on partitions and as such has the ability to infer posterior probabilities over the number of groups in the data. However, the number of data-items, N, is fixed in this case and consistency seems a mute point to worry about. If the prior has parameters that require tuning we can simply use an empirical Bayesian approach.

The situation is different when we train a model on some initial training set of size N_{train} and subsequently test on a larger test set of size N_{test} (note that usually we make independent predictions on test cases in which case we can set $N_{\text{test}} = 1$) . In this case the inferred values of parameters and/or hidden variables on the train set do not necessarily transfer to the larger train+test set. However, if we know in advance the size of the test set (known as "transductive learning") we may define the prior over train+test set but marginalize over the test observations when fitting parameters. In other words, the likelihood terms for the test items are set to 1 and the hidden variables $\mathbf{z}_{\text{test}}$ are carried along as "dummies" and are not connected to actual data. This idea has obvious limitations. In particular it will not work if we don't know how much data we want to test on or when the data arrives in a continuous stream (online learning). Nevertheless, for many applications the extra flexibility may be beneficial.

Why do we want to break away from consistent priors in the first place? Imagine we want to discover segments in an image by clustering groups of pixels. Or imagine we wish to discover topics in a corpus of documents. The DP prior assumes that there are an infinite number of clusters in the world each one with an infinite number of pixels, but with different relative probabilities of being picked. An image is assumed to be a finite random draw of pixels from this infinite set. But do we believe that an image is a random collection of pixels from an infinite image? In particular, if we grow the image larger, do we believe that all clusters will grow larger as well, without bound? Certainly not, because an image may contain many objects entirely and we do not expect them to grow larger as we increase the image size. Similarly, text documents should not be considered as a random draw from an infinite document.

The culprit seems to be that the smaller dataset is not obtained by randomly deleting points from the bigger dataset. Instead, an image is a highly structured subset of the collection of all pixels and finite clusters exist even in the infinite image. These properties violate consistency but more accurately describe the statistics of image segments. The flexible priors that we will propose in this paper are a first attempt to retain some of the convenient properties on nonparametric Bayesian methods (such as inferring the number of groups in the data) but violate consistency in order to gain new modeling flexibility (see [5], [6] for two approaches that violate exchangeability). We have incorporated the flexible prior framework into three well known nonparametric models: the DP mixture model, the HDP and the Beta process (Indian Buffet process) each one of which will be discussed in the following.

II. Flexible Priors over Partitions

The Dirichlet process can be thought of as a prior distribution over all possible *partitions* g of N data-items [5]. A partition is a division of N objects into groups. Objects are considered distinguishable, but clusters are considered indistinguishable. For instance, if we introduce assignment variables z_n for data-item n taking values in the space of cluster labels, we see that the partition $z_1 = 5, z_2 = 5, z_3 = 2$ is equivalent to $z_1 = 3, z_2 = 3, z_3 = 5$, but not equivalent to $z_1 = 5, z_2 = 2, z_3 = 2$. A better notation is therefore $(1, 2), (3)$ to indicate which data-item is in what group.

The simplest derivation of the DP prior over partitions is by taking the infinite limit of a finite mixture model [7] resulting in,

$$p(g) = \frac{\Gamma(\alpha)\alpha^K}{\Gamma(N+\alpha)} \prod_{i=1}^{N} \Gamma(i)^{m_i} \propto e^{\sum_{i=1}^{N}(\log\alpha + \log\Gamma(i))\ m_i} \quad (1)$$

with g indexing partitions, α a fixed parameter related to the expected number of clusters and $\Gamma(\cdot)$ is the Gamma function. Note also that the following relations hold: $K = \sum_i m_i$ and $N = \sum_i i m_i$. The second term expresses the DP prior on partitions as a maximum entropy distribution with parameters $\lambda_i = \log\alpha + \log\Gamma(i)$. The variables $\{m_i\}$ are not independent due to the constraint $\sum_i i m_i = N$.

Distribution 1 is completely specified by the variables $\{m_i\}$ which represent the number of clusters with exactly i data-items in them. Due to this fact, the probability does not change if we exchange the labels of either the clusters or the data-items, i.e. it is exchangeable. Hence, multiple partitions g will have the same probability, since for example the partitions $(1, 2), (3)$ has the same probability as $(1, 3), (2)$. All that counts are the group sizes. Therefore, we can define the "*signature*" as the ordered set of group sizes and represent them by $(xx), (x)$ or simply $(2, 1)$. Signatures are exactly specified by the variables $\{m_i\}$ and they represent the maximal invariants under the permutation group [3]. We can count the number of partitions in a signature to be $N!/\prod_{i=1}^{N}(i!)^{m_i} m_i!$ [8]. Multiplying eqn.1 with this number results in the Ewens sampling distribution.

We will now generalize the DP expression Eq.1 by introducing general features $\phi_a(\mathbf{m})$ which are linearly combined in a log-linear model (or maximum entropy model),

$$\text{FP}(g|N, \boldsymbol{\lambda}) \propto e^{\sum_a \lambda_a \phi_a(\mathbf{m})}\ \mathbb{I}\left(\sum_i i m_i = N\right) \quad (2)$$

Note that eqn.1 is a special case of eqn.2 with $\phi_i(\mathbf{m}) = m_i$ and $\lambda_i = \log(\alpha) + \log\Gamma(i)$. One can also show that the PYP is special case where again $\phi_i(\mathbf{m}) = m_i$ but $\lambda_i = \log\Gamma(i-\gamma) - \log\Gamma(1-\gamma)$ plus one additional nonlinear feature $\phi_0(\mathbf{m}) = \sum_{k=0}^{K(\mathbf{m})-1}\log(\alpha + k\gamma)$ with $K(\mathbf{m}) = \sum_i m_i$, $\lambda_0 = 1$, $\gamma \in [0, 1]$ and $\alpha > -\gamma$.

The FP has N parameters which need to be set. In the following we will discuss two possible methods to do this. The first method converts one's prior assumptions on the average moments $\{\mathbb{E}[\phi_a(\mathbf{m})]\}$ into values for $\{\lambda_a\}$, a process we call "pre-learning" since no data is involved. The second method follows an empirical Bayesian philosophy and learns the parameters directly from data.

A. Pre-learning

We will assume that we have prior knowledge about the expected values of some features over these invariants, i.e. our prior knowledge is expressed as,

$$\mathbb{E}[\phi_a(\mathbf{m})] \approx c_a \qquad a = 1..F \quad (3)$$

Not all choices of c_a are possible due to the constraint on $\mathbf{m}$. The simplest and most practical choice is linear features, $\phi_i(\mathbf{m}) = m_i$, with the constraint that $\sum_i i\mathbb{E}[m_i] = N$. One can also add features to provide information about the variance: $\phi'_i(\mathbf{m}) = m_i^2 - \bar{m}_i^2$ and so on.

There is a danger that due to the discreteness of $\mathbf{m}$ and the constraint $\sum_i i m_i = N$ the specified constraints can not be met exactly. It is therefore safer to state the constraints using a potential function [9] allowing small violations. In this paper we have chosen a simple L_2 norm $U(p) = ||\mathbb{E}_p[\boldsymbol{\phi}] - \mathbf{c}||_2^2/(2\alpha)$. The problem of pre-learning is thus one of solving a maximum entropy problem of the form $\max_p[H(p) - U(p)]$.

The dual formulation is more convenient because it translates into a maximum likelihood problem for the parameters $\boldsymbol{\lambda}$ with weight decay $-\frac{1}{2}\alpha||\boldsymbol{\lambda}||_2^2$. The dual opens the door to more interesting priors on the $\boldsymbol{\lambda}$ parameters. For instance, if we use $\phi_i(\mathbf{m}) = m_i$ then a prior of the form,

$$\log p(\boldsymbol{\lambda}) = A - \alpha||\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{DP}}||_2^2 - \beta\sum_{i=2}^{N}(\lambda_i - \lambda_{i-1})^2 \quad (4)$$

would both keep the values of $\boldsymbol{\lambda}$ close to the DP prior, $\lambda_{i,\text{DP}} = \log(\alpha) + \log\Gamma(i)$ *and* keep the differences between neighboring values of $\boldsymbol{\lambda}$ small (i.e. keep the curve λ as a function i smooth).

Determining values for $\boldsymbol{\lambda}$ is not a standard learning task since there is no real data involved. Rather, we want to convert our prior knowledge expressed as average sufficient statistics into a (maximum entropy) distribution that is approximately consistent with these average sufficient statistics. We call this "pre-learning" to emphasize that no real data is involved.

Pre-learning turns out to be very easy if we have an efficient sampler to sample from $p(g)$. Assuming we have one, pre-learning proceeds by running this sampler and interrupting it at regular intervals to change the parameters $\boldsymbol{\lambda}$ as follows,

$$\lambda_a^{\text{new}} = \lambda_a^{\text{old}} + \eta(c_a - \mathbb{E}[\phi_a(\mathbf{m})]_{\text{FP}} + \nabla_{\lambda_a}\log p(\boldsymbol{\lambda}^{\text{old}})) \quad (5)$$

where η is a learning rate and the average is computed as a sample average from the Gibbs sampler. In theory this method works as long as the rate of change in the parameters is slower than the rate of convergence of the sampler. This algorithm theoretically analyzed in [10] and empirically tested in [11].

Fortunately, Gibbs sampling for the FP in eqn.2 is easy. We will use assignment variables $\{z_n\}$ instead of $\{m_i\}$. Note that Gibbs sampling the $\mathbf{z}$-variables rather than the $\mathbf{m}$-variables ensures that the constraint $\sum_i i m_i = N$ is automatically satisfied. Assume we reassign variable z_m in the current round of Gibbs sampling. We then simply remove that variable from it's current cluster and assign it to all possible existing clusters or to an entirely new cluster. Since we have an explicit expression for the prior in terms of $\mathbf{m}$ and we can compute $\mathbf{m}$ from $\mathbf{z}$ we can easily compute the relative probabilities for each one of these reassignments, $p(z_n = k|z_{\neg n}) \propto \text{FP}(z_n = k, z_{\neg n})$. In the case of features $\phi_i(\mathbf{m}) = m_i$ the prior term can be conveniently expressed as follows. Denote with i the size of the cluster to which item n is currently assigned, and with j the current size of the cluster to which item n will be assigned, then

$$\text{FP}(z_n = k, z_{\neg n}) \propto e^{-\lambda_i + \lambda_{i-1} - \lambda_j + \lambda_{j+1}} \tag{6}$$

B. Empirical Bayes

As an alternative to pre-learning, we can also take an empirical Bayesian point of view and learn hyper-parameters directly from data. In this case one obviously runs a greater risk of over-fitting. It is therefore advisable to parameterize the functional relationship between the $\{\lambda_a\}$ in order to reduce the number of parameters to be learned (see section V) or to use the prior of Eqn. 4 with the values for the hyper-parameters α, β chosen through cross-validation.

The procedure we propose is a simple extension of the procedure described above. The gradient of the data likelihood $\sum_{\mathbf{z}} p(\text{data}|\mathbf{z}) FP(\mathbf{z}|\lambda)$ w.r.t. to the parameters λ_a is given by the difference between two expectations $\mathbb{E}[\phi]$, one over the posterior distribution $p(\mathbf{z}|\text{data})$ and one over the prior $FP(\mathbf{z})$. We thus first sample assignment variables $\{z_n\}$ from the posterior $p(z|\text{data})$. The conditional probabilities are given by,

$$p(z_n = k|z_{\neg n}, \mathbf{x}) \propto p(x_n|x_{\neg n}, z_n = k, z_{\neg n}) \text{FP}(z_n = k, z_{\neg n}) \tag{7}$$

where k is an existing or a new cluster. Note that we included a likelihood term. At regular intervals we interrupt the sampler, record $\{z_n\}$ and continue sampling with the likelihood terms removed (i.e. we sample directly from the prior but initialize at the last iteration from the posterior). Even after a few steps of sampling we will get a good idea of the difference between posterior and prior statistics which we use to perform a learning update on the hyper-parameters, replacing c_a in Eqn.5 with the posterior statistics. This will shrink the prior towards the posterior. After a learning update we recall the last posterior sample and continue sampling from the posterior. The sampler is based on the "contrastive divergence" philosophy [12] where one starts a sampler at the data distribution and relax it for a few iterations towards the model distribution. Here we start at the posterior and let it relax towards the prior.

Although FP does not seem to follow an elegant "Chinese restaurant" (or other culinary) interpretation, its Gibbs sampler is no more or less expensive than the one based on the CRP. The basic reason for this is that exchangeability still holds.

C. Transductive Learning

Let's assume we know beforehand that we will be making predictions on a test set of size N_{test}. In many cases we make predictions independently in which case $N_{\text{test}} = 1$. Because flexible priors are not consistent, a model learned for N_{train} data cases is not necessarily a good model for $N_{\text{train}} + N_{\text{test}}$ data-cases. However, if we know on how many data-cases we will be testing, then we can (pre-) train for $N_{\text{train}} + N_{\text{test}}$ data-cases.

The strategy we propose is to write the probability of the train data by marginalizing over the test data, i.e.

$$\begin{aligned} p(\mathbf{x}_{\text{train}}) &= \sum_{\mathbf{x}_{\text{test}}} \sum_{\mathbf{z}_{\text{train}} \mathbf{z}_{\text{test}}} p(\mathbf{x}_{\text{test}}|\mathbf{z}_{\text{test}}) p(\mathbf{x}_{\text{train}}|\mathbf{z}_{\text{train}}) p(\mathbf{z}_{\text{train}}, \mathbf{z}_{\text{test}}) \\ &= \sum_{\mathbf{z}_{\text{train}} \mathbf{z}_{\text{test}}} p(\mathbf{x}_{\text{train}}|\mathbf{z}_{\text{train}}) p(\mathbf{z}_{\text{train}}, \mathbf{z}_{\text{test}}) \end{aligned} \tag{8}$$

From the last line we see that we can treat the assignment variables $\mathbf{z}_{\text{test}}$ as "dummies" because they are not connected to any likelihood terms and as such are sampled directly from the prior. However, by including them we train a prior that is suitable for $N_{\text{train}} + N_{\text{test}}$ variables in total. This idea effectively circumvents the problem of inconsistency in the transductive learning setting.

The learning algorithm is very similar to the ones described in the previous sections. Now however, our Gibbs sampler reassigns all variables in the set $\{\mathbf{z}_{\text{train}}, \mathbf{z}_{\text{test}}\}$ with $\mathbf{z}_{\text{test}}$ always sampled from the prior $FP(\mathbf{z}_{\text{train}}, \mathbf{z}_{\text{test}})$ (or equivalently with likelihood terms equal to 1). This idea is further developed in section IV for the Hierarchical DP.

III. Illustrative Example

In order to demonstrate learning and applying flexible priors we compare it to the Dirichlet process on a simple clustering problem. One potential drawback of the Dirichlet process is that it can result in many small clusters. Therefore, we learn and then apply a flexible prior that discourages small clusters relative to the Dirichlet process.

In the first step, we specify our prior knowledge through the variables $\mathbf{m}$. In particular, we specify the values for $\mathbb{E}[\mathbf{m}]$ using a truncated normal distribution with $\mu = 0$ and $\sigma = 40$. Relative to the $\mathbb{E}[\mathbf{m}]$ in the Dirichlet process, our distribution for $\mathbf{m}$ puts less strength on the small clusters and more strength on the large clusters (Figure 1-a). Next, we pre-learn the parameters $\boldsymbol{\lambda}$ using the Gibbs sampler described in Section 5. The pre-learned parameters and the parameters for the Dirichlet process are plotted in Figure 1-c.

To compare clustering results, we generate a total of 60 points from three normal distributions, 20 pts each, in 2 dimensions. All clusters have std.=1 and means $\boldsymbol{\mu}_1 = [3,3]$,$\boldsymbol{\mu}_2 =$

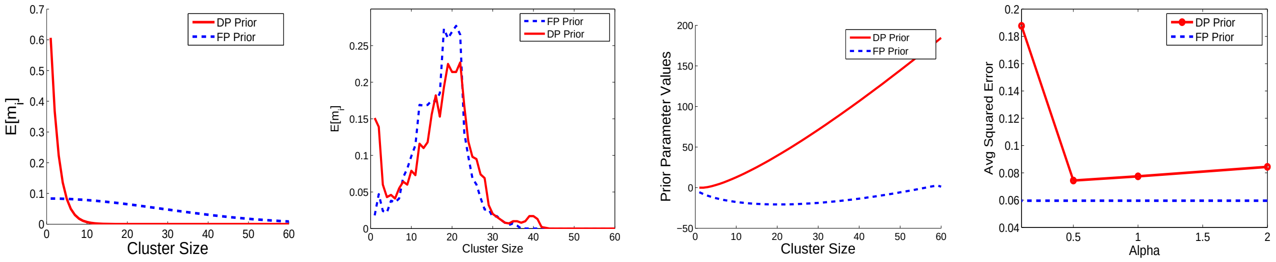

Fig. 1. (a) Prior distribution over $\mathbf{m}$ for DP prior (solid) and FP prior (dashed). (b) Posterior distribution over $\mathbf{m}$ for both models. Note the peak at very small clusters for the DP. (c) Parameter values for $\boldsymbol{\lambda}$ for both models. (d) Error in association between datapoints as a function of α. Note that even for optimal α performance is worse than for the learned prior because small clusters have been suppressed.

$[9, 3]$,$\boldsymbol{\mu}_3 = [6, 6]$. We then cluster these points using a normal-Wishart prior, keeping all hyper parameters the same but varying the prior over partitions. First, four runs of the Gibbs sampler are done with the Dirichlet process prior and four different settings of the α parameter [2,1,.5,.1]. Next, one run is made with the pre-learned prior. For each run, 1000 iterations are used for burn-in and another 1000 iterations are used to calculate the sample mean.

To compare the results we examine two quantities. First, we examine the posterior $\mathbb{E}[\mathbf{m}]$ for the best run using Dirichlet process prior ($\alpha = .5$) versus the run using the learned prior. Figure 1-b shows that as expected the Dirichlet process generates more small clusters than the learned prior. Next, we examine the clustering performance by comparing the average squared difference between the true association between each pair of points and the sampled association. Where the true association between a pair of points = 1 if they were generated by the same cluster and the sampled association = the proportion of samples where the pair of points are assigned to the same cluster. Figure 1-d shows the association error using the Dirichlet process with four different settings for α versus the association error using the learned prior. When α is small, .1, the error is large because the Dirichlet prior encourages only two clusters. As α increases, the Dirichlet process results in more than two clusters, but there are often small clusters sampled with only a few points. Consequently, getting the best results from the Dirichlet process requires careful tuning of the α parameter. It should be noted that if there is enough data, or the data is well separated, using the Dirichlet process prior versus the learned prior has little effect on the posterior results.

IV. Hierarchical Flexible Priors

The HDP [13] extends the DP as a prior for jointly partitioning multiple structured objects (e.g. documents, images) in such a way that elements of different objects (words, pixels) may be assigned to the same global partition (topic), i.e. "topics are shared across documents"[1]. This means that the HDP can be considered as a prior on partitions for all words jointly, but one which is exchangeable only under document-label permutations and permutations of words within documents, but not under permutations of words between documents. The question we now address is whether we can extend the flexible prior framework to this hierarchical setting.

We will need the following notation (see Figure 2). With s_j we will indicate a partitioning of n_j words in document j. Let's call these word-groups. We want these word-groups to be grouped into larger entities which we might call super-groups. Let S be a partitioning of $M \leq N$, $N = \sum_j n_j$ super-groups. We denote a partition at the document level with indicator variables $\{z_{ij}\}$ for word i in document j. For the super-groups we use indicators $\{r_l\}$ which maps a word-group to a supergroup. We need to take special care in mapping the labels in the image of $\mathbf{z}$ and the domain of $\mathbf{r}$. In particular, they need to be consistent and should not create any biases, i.e. we want a random uniform association. We will use the random mapping c to achieve that. Thus, the super-group for word x_{ij} is $r_{c(z_{ij})}$. The full generative process is given by,

$$[z_{1j}, .., z_{n_j j}] \sim \text{FP}_j(\cdot|n_j, \boldsymbol{\beta}_j), \;\; j = 1..D \tag{9}$$

$$[r_1, .., r_M] \sim \text{FP}_0(\cdot|M, \boldsymbol{\alpha}) \tag{10}$$

$$c \sim \text{RP}(\text{co-domain}(\mathbf{z}), M) \tag{11}$$

$$\theta_l \sim \mathcal{G}(\cdot), \;\; l = 1..L \tag{12}$$

$$x_{ij} \sim \mathcal{F}(\cdot|\theta_{r_{c(z_{ij})}}), \;\; i = 1..n_j, \;\; j = 1..D \tag{13}$$

where L is the number of partitions in S that have words assigned to them (i.e. the cardinality of the co-domain of $\mathbf{z}$) and $c|\mathbf{z} \sim \text{RP}(\text{co-domain}(\mathbf{z}), M)$ is a random mapping without replacement from the co-domain of $\mathbf{z}$ to L values in the set $[1, .., M]$ (i.e. randomly permuting the set $[1, .., M]$ and making the first L entries the image of the values in the co-domain of $\mathbf{z}$). Because $L \leq M$ we keep around a number $M - L$ of "dummy" super-groups which are not attached to any word-group and hence not to data.

The HFP thus specifies FPs at the document level and at the topic level. Using an inconsistent FP at the document level is not an issue because we do not expect a document to represent a random subsample from a much larger document. At the topic level we specify our FP over M elements and marginalize out the unnecessary (empty) ones by simply keeping them around as dummies in the sampler. The top level FP can now be used to express our prior expectations over the total number of topics in the corpus.

Gibbs sampling proceeds by sequentially reassigning $\{z_{ij}\}$,

[1] From now on we will simply talk about topics, documents and words.

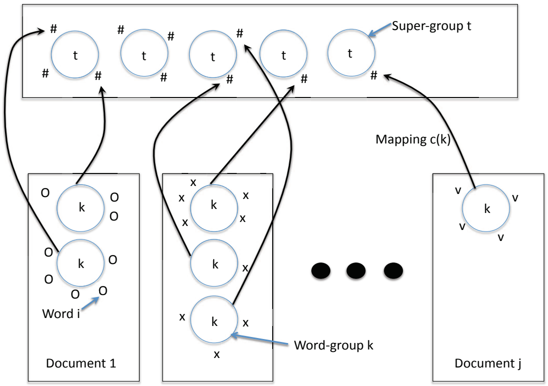

Fig. 2. Graphical representation of HFP model. ”Tables” in the document rectangles represent word-groups. We use $z_{ij} = k$ to denote that word i in document j is assigned to word-group k. Tables at the top rectangle represent super-groups. We use $r_l = t$ to denote the assignment of word-groups to super-groups. Word-groups and super-groups are connected through the random mapping $c(k) = l$. Note the presence of “dummies” at the level of super-groups which are not connected to any data.

$\{r_l\}$ and $\{c(k)\}$ alternatingly. If we reassign z_{ij} we can choose an existing word-group or create a new word-group in document j. If we create a new word-group and label it k_{new}, then we also need to sample a mapping $c(k_{\text{new}})$. Since we condition on the existing mappings $c(k)$ for all instantiated word-groups we must choose the image of $c(k_{\text{new}})$ in the set of dummy super-groups with equal probability $\gamma = 1/(M-L)$. Note that since all dummies assigned to the same super-group have equal probability of being picked we can equally well first compute the probability of picking that super-group proportional to $N_t^{\text{dummy}}/(M{-}L)$ with N_t^{dummy} the total number of dummies in super-group t, and subsequently choose the actual mapping $c(k)$ uniformly at random from that set. The conditional probabilities are thus given by,

$$\textit{If } k \textit{ already exists in doc. } j\textit{:} \quad\quad (14)$$
$$p(z_{ij} = k|-) \propto P_{r_{c(k)}}^{\neg x_{ij}}(x_{ij})\ \mathrm{FP}_j(z_{ij} = k, \mathbf{z}_{\neg ij}|n_j, \boldsymbol{\beta}_j)$$

$$\textit{If } k \textit{ is new:} \quad p(z_{ij} = k, c(k) = l|-) \propto \quad\quad (15)$$
$$\gamma\ P_{r_l}^{\neg x_{ij}}(x_{ij})\ \mathrm{FP}_j(z_{ij} = k, \mathbf{z}_{\neg ij}|n_j, \boldsymbol{\beta}_j)$$

where $\gamma = \frac{1}{M-L}$ is the number of dummy super-groups and

$$P_t^{\neg x_{ij}}(x_{ij}) \propto \int \mathrm{d}\theta\ p(x_{ij}|\theta) \prod_{\substack{i',j' \neq i,j \\ r_{c(z_{ij})}=t}} p(x_{i'j'}|\theta)p(\theta) \quad\quad (16)$$

i.e. we use a collapsed Gibbs sampler which avoids resampling parameters $\boldsymbol{\theta}$.

We will also reassign the super-group assignments, r_l. This will only change the probabilities $\mathrm{FP}_0(M, \boldsymbol{\alpha})$ but not affect $\mathrm{FP}_j(n_j, \boldsymbol{\beta})$. We define $\tilde{c}$ to be the inverse of the mapping c. The conditionals are then given by,

$$p(r_l = t|-) \propto \quad\quad (17)$$
$$P_t^{\neg \mathbf{x}_{\tilde{c}(l)}}(\mathbf{x}_{\tilde{c}(l)})\ \mathrm{FP}_0(r_l = t, \mathbf{r}_{\neg k}|M, \boldsymbol{\alpha})$$

where we note that $\mathbf{x}_{\tilde{c}(l)}$ is the set of all data-items in word-group $k = \tilde{c}(l)$ and

$$P_t^{\neg \mathbf{x}_k}(\mathbf{x}_k) \propto \int \mathrm{d}\theta \prod_{i:z_{ij}=k} p(x_{ij}|\theta) \prod_{\substack{i',j' \neq i,j \\ r_{c(z_{ij})}=t}} p(x_{i'j'}|\theta)p(\theta)$$

Note that in the case l represents a dummy, the inverse mapping $\tilde{c}(l) = \emptyset$. Hence, there will be no likelihood term in Eq.17 implying that dummies are sampled directly from the flexible prior $\mathrm{FP}_0(r_l = t, \mathbf{r}_{\neg k}|M, \boldsymbol{\alpha})$.

To mix over the mappings c, we introduce a swap between two randomly chosen elements of c, i.e. we propose the change $[c(k) = l, c(k') = l'] \rightarrow [c(k) = l', c(k') = l]$. This exchange is equally likely under the prior $p(c)$ and does not change any partitioning structure (i.e. FP_0 and $\mathrm{FP}_j, \forall j$ remain unchanged). However the likelihood terms will be affected resulting in the following MH accept probability,

$$P_{\text{accept}} = \min\left[1, \frac{P_{r_l}^{\neg \mathbf{x}_{k'}}(\mathbf{x}_{k'})P_{r_{l'}}^{\neg \mathbf{x}_k}(\mathbf{x}_k)}{P_{r_l}^{\neg \mathbf{x}_k}(\mathbf{x}_k)P_{r_{l'}}^{\neg \mathbf{x}_{k'}}(\mathbf{x}_{k'})}\right] \quad\quad (18)$$

Again, when l or l' are dummy super-groups there is no pre-image for c and we can set the corresponding likelihood term to 1. (We will not swap between two dummies because it does not change the model.)

Iterating these three sampling updates in succession constitutes our Gibbs sampler for the HFP.

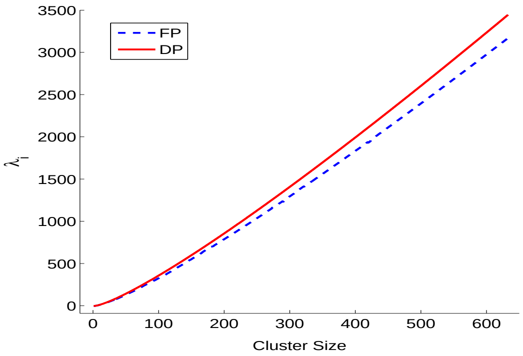

Fig. 3. Resulting $\boldsymbol{\lambda}$ from HFP learning a, b, c in eqn. 19, and HDP learning α.

V. An Experiment with Text Data

In this experiment we compare HFP vs HDP on the KOS text corpus. Our conjecture is that consistency under marginalization is not necessary at the document level when choosing how many words in a document are assigned to a topic, but still makes sense when choosing the topics. That is, we still assume that the topic assigned to each group of words in a

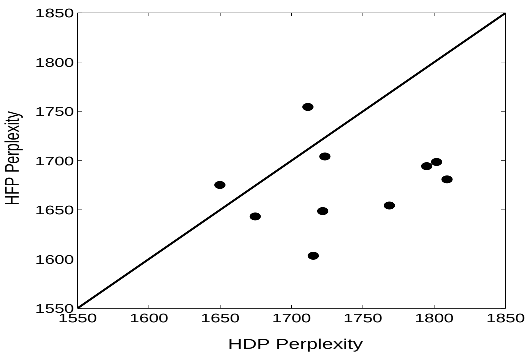

Fig. 4. Comparison of the test perplexity of HDP vs HFP for the KOS text corpus. The x and y coordinates of each point are the perplexity of HDP (x) and HFP (y) on one subset of the KOS text corpus. Points below the diagonal are tests where HFP had lower perplexity than HDP.

document is drawn from an infinite pool of super-topics but we do not believe that a document is a random subset of an infinitely long document (and if it were a subset of a longer article we do not believe that the unobserved remainder is well represented by the same topic distribution). The result is a model where the parameters α of the top level FP$_0$ in equation 10 are set such that the FP is equivalent to a DP, i.e. $\alpha_i = \log(\xi) + \log\Gamma(i)$ (ξ is DP parameter). Inference in the model is done using Gibbs sampling as described in section IV, using the conditional distributions eqn.14,15,17.

To set the values for the parameters $\{\boldsymbol{\lambda}_j\}$ for all flexible priors $\{FP_j\}$ we first assume that the $\boldsymbol{\lambda}$ are not document specific (i.e. do not depend on j) and secondly we use the following parameterized form,

$$\lambda_i = a + b\log\Gamma(i+c) \tag{19}$$

This let's use fit three parameters to data. We use the method described in section II-B based on contrastive divergence to learn these parameters. In the case of the HDP we learn a single parameter α using the same method.

We use the KOS data set [14] for evaluation. We do 10 runs each with $\frac{1}{10}$ of the full data set. For each run we use half of the words in one-quarter of the documents as held out test data and the rest as training data. For each data set we run both HDP and HFP and compare the models in terms of test data perplexity.

To calculate the perplexity we sample the distribution for test words given the current state of the Gibbs chain $\theta_{stj} = \sum_k p(z_{tj} = k|-)P_{r_{c_k}}(x_{tj})$ (tj indexes test word t in document j), where z_{tj} is a latent variable and is sampled as part of the inference procedure, but x_{tj} is unobserved. After allowing the chain to burn-in for 300 iterations, we average over 400 samples $s = 1\ldots S$ to get $\theta_{tj} = \frac{1}{S}\sum_{s=1}^{S}\theta_{stj}$. Perplexity is then $\exp(-1/T\sum_{tj}\log p(x_{tj}|\theta_{tj}))$. For HDP we make the equivalent calculation using samples from the Chinese restaurant franchise sampling scheme [15].

Figure 4 compares the perplexity of HFP and HDP on 10 subsets of KOS. We find a small but significant improvement in terms of perplexity between HDP and HFP. The average perplexity for HDP and HFP is 1737 and 1675, respectively. Using a paired t-test we can conclude with a p-value of 0.01 that the differences are significant. We can also examine the difference between the learned $\boldsymbol{\lambda}$ for HFP and HDP (see figure 3).

These experiments represent preliminary evidence that we can learn better nonparametric models by relaxing consistency and that our proposed inference procedures work properly.

VI. Flexible Priors for Binary Matrices

We now extend flexible priors to binary matrices. Our starting point is the "Indian Buffet Process" (IBP) introduced in [7]. Relative to DPs, we relax the requirement that a data-case is assigned to a single cluster only. As in [7] we will use the "left-ordered-form" representation of binary matrices, H, ordering the columns by their binary representation. Analogous to priors on partitions, we can write the "Indian buffet process" (IBP) prior from [7] as a maximum entropy distribution as follows,

$$\begin{aligned} p(H) &= \frac{\alpha^K}{\prod_{h=1}^{2^N-1} K_h!} e^{-\alpha H_N} \prod_{i=1}^{N}\left(\frac{\Gamma(N-i+1)\Gamma(i)}{\Gamma(N+1)}\right)^{m_i} \\ &\propto \frac{1}{\prod_{h=1}^{2^N-1} K_h!} e^{\sum_i m_i(\log\alpha + \log\Gamma(N-i+1) + \log\Gamma(i) - \log\Gamma(N+1))} \end{aligned} \tag{20}$$

where $H_N = \sum_{i=1}^{N} 1/i$, K_h is the number columns with binary representation h, m_i the number of features that is being used by i items, and $K = \sum_i m_i$ the total number of features. Note that $\prod_h K_h!$ is also invariant under permutations of rows and columns. We will include this factor in the generalized priors because it happens to simplify the MCMC samplers. Also note that as in the case of partitions $g = g(\mathbf{m})$, there will be multiple left-ordered binary matrices $H = H(\mathbf{m})$ with the same probability that correspond to a single setting of $\mathbf{m}$.

To generalize the IBP prior we first find a suitable set of invariants. Clearly, the $\{m_i\}$ variables are still invariant under permutations of rows and columns of a binary matrix. However, there is another set of useful invariants which we will indicate with $\{n_j\}$. These represent the number of objects that are member of exactly j clusters. Note that for partitions, where every data-case can only be member of exactly one group, this invariant reduces to $n_1 = N, n_j = 0,\ j > 1$ and is therefore uninformative. The $\mathbf{m}$ and $\mathbf{n}$ variables satisfy two important constraints,

$$\sum_{i=1}^{N} i m_i = \sum_{j=1}^{\infty} j n_j \qquad \sum_{j=1}^{\infty} n_j = N \tag{21}$$

Binary matrices may be viewed as bipartite graphs with a potentially infinite number of top layer nodes and exactly N bottom layer modes. While m_i represents the number of top-layer nodes with i outgoing edges, n_j represents the number of bottom layer nodes with j incoming edges.

We define features $\phi_a(\mathbf{m},\mathbf{n})$ and the maximum entropy distribution,

$$FP(H|N,\boldsymbol{\lambda}) \propto \frac{1}{\prod_{h=1}^{2^N-1} K_h!} e^{\sum_{a=1}^{F} \lambda_a \phi_a(\mathbf{m},\mathbf{n})} \times \\ \mathbb{I}\left(\sum_i i m_i = \sum_j j n_j \wedge \sum_j n_j = N\right) \tag{22}$$

The IBP prior is a special case of this more general prior when we set $\phi_i(\mathbf{m}) = m_i$, $\lambda_i = \log\alpha + \log\Gamma(N-i+1) + \log\Gamma(i) - \log\Gamma(N+1)$.

Determining an appropriate setting for the parameters $\boldsymbol{\lambda}$ may again be achieved by specifying the average feature values consistent with the constraints and updating the parameter values with equations similar to eqn.5 inside an MCMC algorithm that samples from eqn.22. As an alternative we can again reduce the number of parameters by assuming some functional form and using the empirical Bayesian method explained in section II-B to fit the remaining parameters.

The practicality of these new priors depends on the existence of efficient samplers. In the next section we will discuss two such samplers.

VII. MCMC Samplers for FP-IBP

We will first generalize the IBP sampling algorithm to priors of the form 22 where we set $\phi_i(\mathbf{m},\mathbf{n}) = \phi_i(\mathbf{m}) = m_i,\ i = 1..N$ but the values for $\boldsymbol{\lambda}$ are arbitrary. In this case we can use a Gibbs sampler that is a direct generalization of the one proposed in [7]. We define z_{kn} as the binary indicator for whether item n is assigned to feature k, and $N_k^{\neg n} = \sum_{m \backslash n} z_{km}$ as the total number of items assigned to feature k not counting item n. Also let $\sigma(\cdot)$ be the sigmoid function. The following Gibbs sampler then samples from the flexible IBP prior. Due to the combinatorial factor $\prod_h K_h!$, the values for z_{kn} can be Gibbs sampled *independently*. It exactly cancels against counting factors generated by the fact that we work with left-ordered-form binary matrices [7].

Algorithm 1 MH-MCMC Sampler 1 for Flexible Priors on Infinite Binary Matrices

```
    Repeat:
 1: For $n = 1$ to $N$
 2: For $k = 1$ to $K$
2a: If $N_k^{\neg n} = 0$ delete feature.
2b: If $N_k^{\neg n} > 0$ do
2c: With probability $p = \sigma(\lambda_{N_{\texttt{new}}} - \lambda_{N_{\texttt{old}}})$, where $N_{\texttt{new}} = N_k^{\neg n} + \neg z_{kn}$ and $N_{\texttt{old}} = N_k^{\neg n} + z_{kn}$, set $z_{kn}$ to 1.
3a: Sample $N_+$ from a Poisson distribution Pois$(e^{\lambda_1})$.
3b: Set $z_{k+1,n}, ..., z_{k+N_+,n}$ to 1 and $\mathbf{z}_{k+1,\neg n}, ..., \mathbf{z}_{k+N_+,\neg n}$ to 0 for all other data-cases.
```

Unfortunately, this sampler does not generalize to priors of the general form in eqn.22. It is however not too difficult to derive a Metropolis-Hastings (MH) sampler for the general FP-IBP. Below we describe one such example that worked well enough in our experiments but better versions with improved mixing behavior are likely to exist. The basic idea is to alternate performing MH updates on the existing K clusters of the binary matrix with adding a single new feature with a single new item in it. When we pick an item in an existing cluster one of two things can happen: either flipping the bit in the binary matrix removes the last object from the feature or it does not not. If not, the update is basically a simple Gibbs update. If the feature is emptied then we need to make sure detailed balance is maintained by incorporating the correct acceptance probability. The sampler below can be shown to satisfy detailed balance. (In the algorithm box below, we use: $f(\mathbf{m},\mathbf{n}) \doteq \sum_{a=1}^{N} \lambda_a \phi_a(\mathbf{m},\mathbf{n})$)

Algorithm 2 MH-MCMC Sampler 2 for Flexible Priors on Infinite Binary Matrices

```
    Repeat:
 1: With probability $\frac{K}{K+1}$ perform an update on one of the first $K$ (existing) clusters.
1a: Choose a bit $z_{kn},\ k < K$ uniformly at random.
1b: If $N_k^{\neg n} > 0$ propose to flip the bit.
1c: Accept move with probability: $p_{\texttt{acc}} = \min[1, \exp(f(\mathbf{m}^{\texttt{new}}, \mathbf{n}^{\texttt{new}}) - f(\mathbf{m}^{\texttt{old}}, \mathbf{n}^{\texttt{old}}))]$.
1d: If $N_k^{\neg n} = 0$ propose to remove the feature.
1e: Accept move with probability $p_{\texttt{acc}} = \min[1, \frac{K+1}{K} \exp(f(\mathbf{m}^{\texttt{new}}, \mathbf{n}^{\texttt{new}}) - f(\mathbf{m}^{\texttt{old}}, \mathbf{n}^{\texttt{old}}))]$.
 2: With probability $\frac{1}{K+1}$ propose to add a new feature to the matrix.
2a: Sample an item $n$ uniformly at random.
2b: Set $z_{K+1,n} = 1$ and $z_{K+1,\neg n} = 0$.
2c: Accept move with probability $p_{\texttt{acc}} = \min[1, \frac{K+1}{K+2} \exp(f(\mathbf{m}^{\texttt{new}}, \mathbf{n}^{\texttt{new}}) - f(\mathbf{m}^{\texttt{old}}, \mathbf{n}^{\texttt{old}}))]$.
```

Note that the computation for $\Delta = f(\mathbf{m}^{\texttt{new}}, \mathbf{n}^{\texttt{new}}) - f(\mathbf{m}^{\texttt{old}}, \mathbf{n}^{\texttt{old}})$ simplifies considerably if we use linear features. In that case we get: $\Delta = \lambda_{N_{\texttt{new}}} - \lambda_{N_{\texttt{old}}} + \theta_{F_{\texttt{new}}} - \theta_{F_{\texttt{old}}}$ where $N_{\texttt{new}} = N_k^{\neg n} + \neg z_{kn}$, $N_{\texttt{old}} = N_k^{\neg n} + z_{kn}$, $F_{\texttt{new}} = \sum_{j \backslash k} z_{jn} + \neg z_{kn}$ and $F_{\texttt{old}} = \sum_{j \backslash k} z_{jn} + z_{kn}$. We also note that sampling from the posterior will require some extra likelihood terms in the acceptance probability. We have tested these samplers on a simple problem and found empirically that they mix reasonably well. Some results are reported in the next section.

VIII. Evaluation of the FP-IBP MCMC Sampler

The usefulness of the new family of priors crucially depends on the availability of a good sampling algorithm. In this section we will compare the mixing properties of these samplers on the "noisy-or" problem proposed in [16]. In this problem $\mathbf{z}$ determines the structure of a two-layer Bayes net. The binary unobserved states of the top layer, $\mathbf{y}$, are independently distributed a priori according to a Bernoulli distribution,

$$p(\mathbf{y}) = \prod_{k,t} p^{y_{k,t}} (1-p)^{1-y_{k,t}} \tag{23}$$

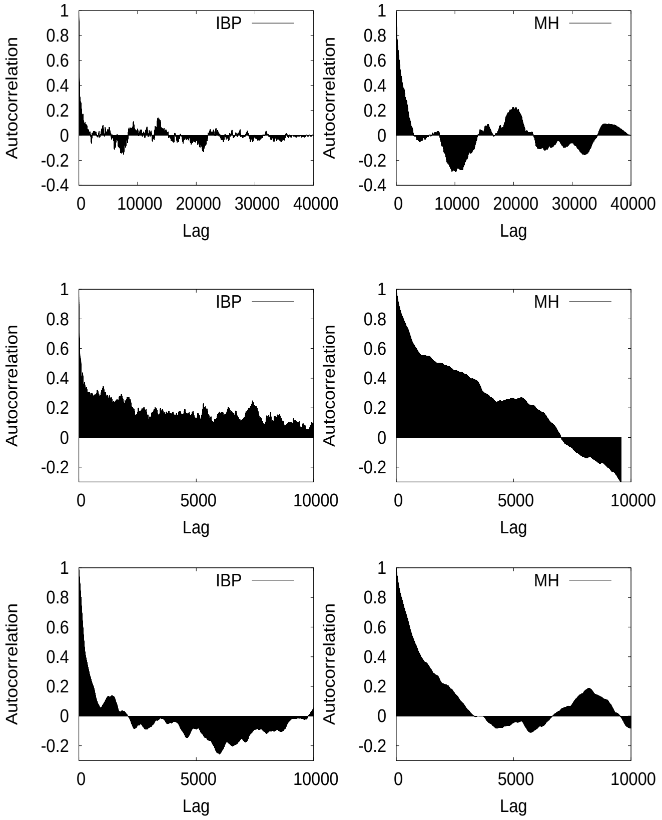

Fig. 5. Autocorrelation plots for IBP sampler (left) and MH sampler (right). First row: $N = 10$, $K = 8$, $\alpha = 2$, $T = 50$. Second row: $N = 5$, $K = 16$, $\alpha = 3$, $T = 60$. Third row: $N = 15$, $K = 8$, $\alpha = 3$, $T = 20$. Likelihood parameters: $\varepsilon = 0.01$, $\lambda = 0.9$, $p = 0.1$, see [16].

where t indexes the data-item. The observed states of the bottom layer are distributed according to a “noisy-or” model,

$$p(x_{i,t} = 1|\mathbf{y}, \mathbf{z}) = 1 - (1 - \lambda)^{[\mathbf{z}^T\mathbf{y}]_{i,t}}(1 - \varepsilon) \qquad (24)$$

We have used IBP priors for $\mathbf{z}$.

We have implemented the Gibbs sampler proposed in [16] to sample from the posterior. Sampling from the posterior with our MH sampler needs a small change in the acceptance rule. If we sample a new value for $\mathbf{y}$ from the prior whenever we need it, then all acceptance ratios in section VII are multiplied by the ratio $p(\mathbf{x}|\mathbf{y}^{\texttt{new}}, \mathbf{z}^{\texttt{new}})/p(\mathbf{x}|\mathbf{y}^{\texttt{old}}, \mathbf{z}^{\texttt{old}})$.

We note that the moves for existing clusters are very similar between the IBP and MH sampler, but that the moves which change K are different. Hence, we have compared the auto-correlation function for the number of clusters K. We collect a sample after each update on a pair $(z_{i,k}, y_{k',t})$. For the IBP sampler we count the generation of N_+ new clusters from a Poisson distribution including their new y values also as a single sample. Plots of auto-correlation functions computed after a long burn-in phase for various settings of N, T, K and α are shown in figure 5.

From these plots we see that the IBP sampler mixes faster, by about a factor of $2 - 4$. We emphasize however that the MH sampler is much more general than the IBP sampler since it applies to general nonlinear features for both $\mathbf{m}$ and $\mathbf{n}$ variables.

IX. Discussion

We have argued that the usual consistency constraint for nonparametric models may not be necessary and indeed sometimes even be inappropriate. In light of that, we have proposed a new family of *flexible priors* which may violate consistency but still satisfy exchangeability. The latter property is a necessary ingredient for efficient inference (besides often being an appropriate assumption to make). We have specified procedures to set parameters of these models by either pre-learning from prior knowledge or by directly learning from data. The flexible prior framework is quite general as evidenced by the fact that we were able to formulate FP extensions for the DP, HDP and IBP models and provide efficient inference procedures. Preliminary experiments provide evidence that we can successfully learn these models from data.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 0447903, 0914783, 0928427.

References

[1] S. MacEachern and P. Müller, “Estimating mixture of Dirichlet process models,” *Communications in Statistics*, vol. 7, pp. 223–238, 1998.

[2] R. Neal, “Markov chain sampling methods for Dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, pp. 283–297, 2000.

[3] P. Green and S. Richardson, “Modelling heterogeneity with and without the Dirichlet process,” *Scandinavian Journal of Statistics*, vol. 28, 2001.

[4] J. Pitman, *Combinatorial Stochastic Processes*. Berlin: Springer-Verlag, 2006, available at: http://works.bepress.com/jim-pitman/1 and via SpringerLink.

[5] F. Quintana and P. Iglesias, “Bayesian clustering and product partition models,” *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 65, pp. 557–574, 2003.

[6] H. Wallach, S. Jensen, L. Dicker, and K. Heller, “An alternative prior process for nonparametric bayesian clustering,” in *International Workshop on Artificial Intelligence and Statistics*, Y. Teh and M. Titterington, Eds., vol. 9, 2010, pp. 892–899.

[7] T. Griffiths and Z. Ghahramani, “Infinite latent feature models and the indian buffet process,” in *Advances in Neural Information Processing Systems 18*, 2006, pp. 475–482.

[8] C. Antoniak, “Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems,” *The Annals of Statistics*, vol. 2, pp. 1152–1174, 1974.

[9] M. Dudík and R. E. Schapire, “Maximum entropy distribution estimation with generalized regularization,” in *COLT*, 2006, pp. 123–138.

[10] L. Younes, “Parametric inference for imperfectly observed gibbsian fields,” *Probability Theory and Related Fields*, vol. 82, 1989.

[11] T. Tieleman, “Training restricted boltzmann machines using approximations to the likelihood gradient,” in *Proceedings of the International Conference on Machine Learning*, vol. 25, 2008, pp. 1064–1071.

[12] G. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, pp. 1771–1800, 2002.

[13] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” *To appear in Journal of the American Statistical Association*, 2006.

[14] A. Frank and A. Asuncion, “UCI machine learning repository,” 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[15] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Hierarchical Dirichlet processes,” in *NIPS*, vol. 17, 2004.

[16] F. Wood, T. L. Griffiths, and Z. Ghahramani, “A non-parametric bayesian method for inferring hidden causes,” in *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, 2006, pp. 536–543.