Flexible Priors for Infinite Mixture Models

Max Welling WELLING@ICS.UCI.EDU
Bren School of Information and Computer Science, University of California Irvine, CA 92697-3425 USA

Abstract

Most infinite mixture models in the current literature are based on the Dirichlet process prior. This prior on partitions implies a very specific (a priori) distribution on cluster sizes. A slightly more general prior known as the Pitman-Yor process prior generalizes this to a two-parameter family. The latter is the most general exchangeable partition probability function (EPPF) as defined by Pitman (Pitman, 2002) known to date. I want to argue that it is desirable to have more flexibility in expressing our prior beliefs over cluster sizes. EPPFs as defined by Pitman satisfy 3 conditions, exchangeability over objects, exchangeability over cluster-labels and consistency. In this contribution I explore the possibility to relax some of these conditions. In particular, I will discuss the consequences of relaxing exchangeability over cluster-labels and consistency. In both cases it turns out that one can formulate proper and efficient Gibbs sampling algorithms but with the added flexibility of having more control to design one's prior.

1. Preliminaries

Dirichlet process mixture models (Ferguson, 1973; Blackwell & MacQueen, 1973) represent an elegant solution to clustering data with the added benefit of inferring a sensible number of clusters fully automatically. In fact, like other non-parametric Bayesian models, the complexity of DP mixtures models grows with the availability of more data. This property is appropriate for many real world data problems, where it is unrealistic to assume that the data was sampled form a model with a finite number of parameters. This property of growing model complexity has made the term "infinite models" popular in machine learning.

I will use a general setup, where $X = \{X_{in}\}$ represents the data matrix with i denoting the attribute index and n the data-item index. For clustering we also need a assignment variable $Z = \{Z_n\}$ who's value is the cluster label to which data-item n belongs. Furthermore, depending on the likelihood model we need some parameters such as the mean and the covariance which will be denoted by θ. However, here we will not concern ourselves with those details. The joint model over X, Z is given by,

$$p(X, Z) = p(X|Z)P(Z) \tag{1}$$

The first term is actually an integral over all possible parameters weighted by the prior over those parameters (e.g. a normal-Wishart distribution for the mean and inverse covariance),

$$p(X|Z) = \int d\theta P(X|Z, \theta)p(\theta) \tag{2}$$

For conjugate priors these integrals can indeed be performed analytically.

The DP clustering algorithm is based on a Gibbs sampler and is impressively simple (Escobar & West, 1995; Bush & MacEachern, 1996; MacEachern & Müller, 1998; Neal, 2000; Green & Richardson, 2001). We cycle through the data-items, sampling their assignment variable according to the posterior distribution

$$p(Z_n = k|Z_{\neg n}, X) \propto p(Z_n = k|Z_{\neg n}) \times \\ \times \int d\theta_k p(X_n|Z_n, \theta_k)p(\theta|Z_{\neg n}, X^k_{\neg n}) \tag{3}$$

where $X^k_{\neg n}$ indicates the subset of data which is currently assigned to cluster k, taking particle n out of the sample and where,

$$p(Z_n|Z_{\neg n}) = \frac{\sum_k N_k^{\neg n}\delta_{z_n,k} + \alpha\delta_{z_n,K+1}}{N - 1 + \alpha} \tag{4}$$

with $\delta_{i,j}$ the delta-function which is zero everywhere except when $i = j$. The second part in eq.(3) is thus the probability of the particle under the predictive distribution of cluster model k (leaving particle n out of the calculation of the posterior distribution of the parameters θ). We will not be concerned with this term as it is the same for all methods that we will consider in the following. Instead we will focus on the specification of the prior $p(Z)$.

Appearing in *Proceedings of the Workshop on Learning with Nonparametric Bayesian Methods*, 23^{rd} ICML, Pittsburgh, PA, 2006.

2. Exchangeable Partition Probability Functions (EPPF)

The prior $p(Z)$ for a DP is given by a variant of the Ewen's sampling formula,

$$p(N_1,..,N_K) = \frac{\alpha^K \Gamma(\alpha)}{\Gamma(\alpha+N)} \prod_{i=1}^{K} (N_i - 1)! \tag{5}$$

An explicit formula for the more general Pitman-Yor (PY) process also exists (Pitman, 2002).

EPPFs are defined by Pitman to satisfy 3 fundamental properties,

1. Exchangeability over particles. This is achieved in eq.(5) by the fact that it is expressed in terms of invariants w.r.t. changing the order of the particles, namely N_i,N and K, with N_i the number of particles in cluster i, N the total number of particles and K the number of occupied clusters respectively.

2. Exchangeability over cluster labels. This is achieved in eq.(5) by making sure the EPPF is symmetric under permutations of the cluster labels i. Instead, one could order the cluster counts N_i according to their size and express the EPPF as a function of these ordered counts.

3. Consistency. A consistent EPPF follows the following addition rule for probabilities:

$$\begin{aligned} p(N_1,..,N_K) = \sum_{i=1}^{K} p(N_1+\delta_{i,1},..,N_K+\delta_{i,K}) \\ + p(N_1,..N_K,1) \end{aligned} \tag{6}$$

   One could think of the process of creating a new particle as a physical process, where the total probability of creating a new particle from an old state, must be the sum of the probabilities of all possible ways to create that particle. For example, $p([1,2]) = p([1,2,3]) + p([1,2],[3])$ and $p([1],[2]) = p([1,3],[2]) + p([1],[2,3]) + p([1],[2],[3])$, where the notation e.g. $[1,2],[3]$ is used to denote that particle 1 and 2 are in the same cluster and particle 3 is in a separate cluster.

An EPPF can also be derived as the limit of a finite Dirichlet process, where we send the number of clusters (including clusters with zero particles) to infinity. In this procedure one needs to include the proper counting factors to make sure one makes the transition between explicit cluster-labels to equivalence classes over cluster-labels (or partitions) where many equivalent labelings are collapsed on one state (Griffiths & Ghahramani, 2006). This type of derivation enforces all three properties automatically. Exchangeability over particles is observed because the representation $p(Z) = \int d\theta \prod_n p(Z_n|\theta)p(\theta)$ makes this explicit. Exchangeability over cluster labels is observed because the construction is in the space of equivalence classes over all equivalent labelings. Finally, consistency is observed because for arbitrary K we have

$$\begin{aligned} \int d\theta \prod_{n=1}^{N-1} p(Z_n|\theta)p(\theta) = \\ \sum_{z_N=1}^{K} \int d\theta p(Z_N|\theta) \prod_{n=1}^{N-1} p(Z_n|\theta)p(\theta) \end{aligned} \tag{7}$$

since $\sum_{z_N=1}^{K} p(Z_N|\theta) = 1$.

3. Relaxing Conditions

The most general family of EPPFs known (to the best of my knowledge) is given by the two-parameter family of Pitman-Yor processes which includes the DP (Pitman, 2002). This family of priors implies certain properties on the distribution of prior cluster sizes. More specifically, consider an infinite, size-ordered sequence of probabilities $\pi_1 \geq \pi_2, ...$ Define a size-biased random permutation of this sequence by sampling indices sequentially from $\{\pi_i\}$ without replacement, i.e. if we sampled index j according to $\{\pi_i\}$, then we remove π_j from the pool and normalize. One can show that the distribution of average cluster sizes for a DP(α) according to a size-biased ordering of the clusters decays exponentially with decay rate governed by α (Pitman, 2002),

$$\langle n_i \rangle = \frac{N}{\alpha} e^{-i \log\left(\frac{1+\alpha}{\alpha}\right)} \tag{8}$$

For certain parameter settings of the Pitman-Yor process the tails can be made to behave according to a power-law[1].

The heavy tails of these distributions are somewhat restrictive and may not accurately express our prior expectation about cluster size distributions. Hence, we study the consequences of relaxing some of the conditions below.

3.1. (Not) Relaxing Particle Exchangeability

We start by defining what we mean by a partition. A partition is defined to be a division of objects into clusters. In a partition we do not care about the way the clusters are labelled, i.e. $Z_1 = 1, Z_2 = 1, Z_3 = 2$ is identical to $Z_1 = 5, Z_2 = 5, Z_3 = 1$. A better notation is to use brackets $[1,2],[3]$ denoting $Z_1 = Z_2 \neq Z_3$.

Exchangeability over particles means that the probability of partitions with the same "signature" is equal. A

[1] In fact, one can show that the asymptotic frequencies are distributed according to the stick-breaking process defined in (9).

signature is the size ordered sequence of count vectors $N_1 \geq N_2 \geq ... \geq N_K$. In our example: $p([1,2],[3]) = p([1,3],[2]) = p([2,3],[1])$. One can count the number of partitions with equal signature using the definition m_j which is the number of clusters of size j, i.e. we have $\sum_{j=1}^{N} j m_j = \sum_{i=1}^{K} N_i = N$ and $\sum_{j=1}^{N} m_j = K$. In terms of these, the number of partitions with signature $N_1 \geq N_2 \geq ... \geq N_K$ is $\frac{N!}{\prod_{j=1}^{N}(j!)^{m_j} m_j!}$ which represents all permutations of the available particles and correcting for over-counting the permutations among particles in the same cluster and the exchange of all particles between two clusters of the same size.

Exchangeability over particles is a very natural requirement for a prior. In most cases we have no information that particle 5 is more likely to be in a big cluster, or less likely to be in the same cluster as particle 20. The likelihood term however, is *not* exchangeable over particles. Indeed, clustering can precisely be interpreted as making the right associations between particles, and hence it can only be a property of the prior. Relaxing it therefore seems a very bad idea; if we allow non-exchangeable priors we will not be expressing our ignorance in this respect and bias our clustering algorithm accordingly. Moreover, it is really easy to avoid it: simply express your prior in terms of counts $\{N_i\}$ only.

3.2. Relaxing Cluster Label Exchangeability

The Gibbs sampler for the DP model samples in the space of *equivalence classes* over cluster labels, i.e. we consider the space of partitions only and treat a relabelling of the clusters as indistinguishable. This is possible because both the prior and the likelihood terms are invariant w.r.t. permutations of the cluster labels implying that cluster labels are unidentifiable.

What will happen then, if we relax the requirement that the prior is invariant under these cluster relabelings? Interestingly, the well known relationship between stick-breaking priors and the DP does exactly this (Sethuraman, 1994). This relationship provides a precise expression for the limiting values of relative cluster sizes according to the size-biased ordering of clusters. This also happens to be the ordering that one would obtain if one samples from the DP prior according to the Chinese restaurant process.

In the stick-breaking representation we represent a prior $p(Z)$ by breaking a virtual stick infinitely often where the break point is determined by drawing a beta variable with parameters $a_k = 1, b_k = \alpha$. The length of the (say) left remainder is denoted V_1. We then break the right remainder again using a new beta variable and call the left of this remainder V_2 etcetera ad infinitum. The prior probabilities of the clusters are then combined via the rule

$$\pi_k = V_k \prod_{j<k}(1 - V_j) \tag{9}$$

This construction immediately provides an opportunity to generalize the DP, by picking arbitrary $\{a_k, b_k\}$. For instance, the PY process is obtained by $a_i = 1 - a$ and $b_i = b + i \times a$ for $a \in [0, 1)$ and $b > -a$.

It is now not hard to derive the conditional distributions $p(Z_n|Z_{\neg n})$ which are needed for the Gibbs sampler, where we have marginalized out the beta variables (Ishwaran & James, 2001; Ishwaran & James, 2003),

$$p(Z_n = k|Z_{\neg n}) = \frac{a_k^*}{a_k^* + b_k^*} \prod_{j<k} \frac{b_j^*}{a_j^* + b_j^*} \tag{10}$$

where $a_i^* = a_i + N_i^{\neg n}$, $b_i^* = b_i + \sum_{j=i+1}^{\infty} N_j^{\neg n}$ and $N_i^{\neg n} = \sum_{n' \backslash n} \delta_{z_{n'},i}$ is the number of particles in cluster i. Finally we construct the prior $p(Z)$ by adding the particles sequentially, (Ishwaran & James, 2003),

$$p(Z) = p(Z_1) \prod_{n=2}^{N} p(Z_n|Z_1, .., Z_{n-1}) \tag{11}$$

where we have used some arbitrary ordering of the particles. Note that this prior is exchangeable over particles because it was derived by marginalizing out conditionally independent beta variables, and hence the result does not depend on the ordering of the data. However, the prior is *not* generally exchangeable over cluster labels. This fact can be more easily seen if we choose a special case of the above more general result where $a_i = \gamma_i$ and $b_i = \sum_{j=i+1}^{\infty} \gamma_i$ (Ishwaran & James, 2003). In that case we have $a_i + b_i = b_{i-1}$ resulting in,

$$p(Z_n = k|Z_{\neg n}) = \frac{\gamma_k + N_k^{\neg n}}{\gamma + N - 1} \qquad \gamma = \sum_{i=1}^{\infty} \gamma_i \tag{12}$$

and we use again eq.(11) to combine this into $p(Z)$ using some ordering of the particles. We clearly see that the clusters have different a priori probabilities to have a certain size, where the first cluster is expected to be largest etc. The result is that the Gibbs sampler using the above prediction rule, which looks deceptively similar to the one for the DP, will *not* sample in the space of partitions, but rather in the space of explicit cluster labels. In other words, unlike the DP, the posterior probability $P(Z|X)$ has an exponential number of modes created by permutations of cluster labels which are *not* equivalent: if we swap cluster labels the probability of the new configuration is different. How can we reconcile this with the fact that labels are unidentifiable? The answer is that the probability of a particular clustering of the data is the sum over all possible labelings of the clusters. The Gibbs sampler however is likely to explore only one (or perhaps a few) of those modes assigning a potentially biased probability to that clustering. It is important to reiterate that the DP does not suffer from this problem because relabeling results in equivalent probabilities, so we only have to explore a single representative mode (i.e. we sample in the space of equivalence classes or partitions).

To drive this point home, let's consider sampling two data assignments, both in the same cluster. For a DP we know from the prediction rule that the total probability of this event is $1/(1+\alpha)$. However, in the stick breaking representation we have to sum over all labels to obtain that same result,

$$P(z_1 = z_2) = \sum_{i=1}^{\infty} P(z_2 = i | z_1 = i) P(z_1 = i) \qquad (13)$$

which after some algebra can be found to be equal to $1/(1+\alpha)$ as well.

To illustrate the effect of poor mixing between cluster labels, we generated the symmetric dataset in figure (1). We use a standard normal-Wishart prior centered at zero and tuned so that most of the time two clusters best explain the data. The central data-case should be assigned with probability 0.5 to the left or the right cluster, due to symmetry. To test this we used a prior on cluster sizes with $a_i = 5$ and $b_i = 0.1$ in eq. (10). We then run the Gibbs sampler for 5000 iterations (discarding the first 100 for burn-in). We measure the average association between the central data-case and all other data-cases, where two data-cases are associated if they have the same label. The results are shown in Figure (2). Clearly, the prior favors association with one cluster (right block of 25 data-cases) over the other, but due to symmetry this should not be the case. Poor mixing between the many extra modes introduced by sampling in the space of cluster labels instead of partitions can therefore lead to biased estimates[2]

One can fix the illustrated problem by introducing extra mixing moves which swap the cluster labels and accept/reject them using Metropolis-Hastings rules. The moves have the highest probability of swapping between dominant modes. We have observed that this removes the cluster bias in the above example (Porteous et al., 2006). Note that the variational approximation in the stick-breaking representation (Blei & Jordan, 2005) is prone to the same phenomenon. However, the effect is usually very small due the fact the likelihood term is much stronger than the prior term.

Consistency can also be defined in a meaningful way for the stick-breaking prior. We require that,

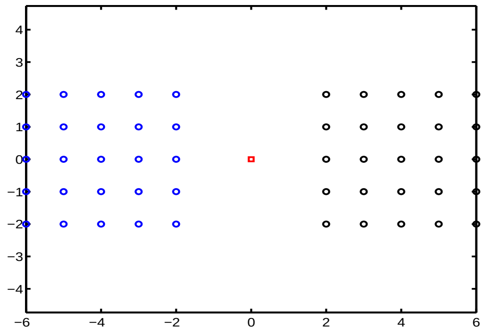

Figure 1. The symmetric data-set to illustrate clustering bias in the absence of mixing moves.

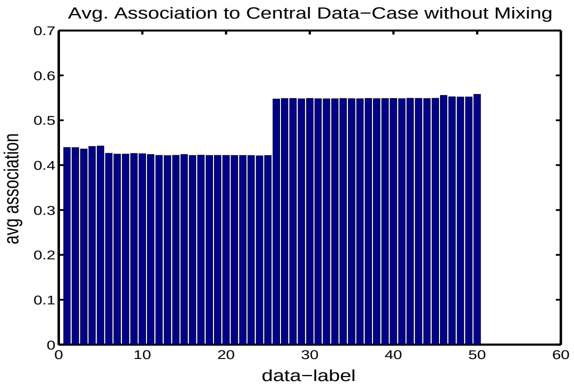

Figure 2. Average association of center point to all other data-cases for a sampler without mixing moves. Note the cluster bias.

$$p(N_1, .., N_\infty) = \sum_{i=1}^{\infty} p(N_1 + \delta_{1,i}, N_2 + \delta_{2,i}, ..) \qquad (14)$$

where we included an infinitely large set of empty clusters with $N_k = 0$. Due to the way $p(Z)$ is constructed using eq. (11) consistency is trivially preserved. A similar argument as the one around eq.(7) can also be used to show this.

In conclusion we view relaxing the exchangeability over cluster labels as a trade-off between the convenience of sampling in the space of equivalence classes versus the added flexibility of designing one's prior.

3.3. Relaxing Consistency

In the appendix of (Green & Richardson, 2001) it is briefly mentioned that consistency may not be essential for clustering but that it is certainly a very constraining requirement in the design of priors. As our purpose is to free ourselves from the most constraining conditions in order to gain flex-

[2] Note that the sampler is not "wrong" in a theoretical sense, it just mixes poorly.

ibility in constructing our priors, this seems a very promising route to that goal.

First let's consider an example of an exchangeable probability distribution on partitions (i.e. exchangeable both over particles and cluster labels) that does not satisfy consistency. We'll use the simplest possible distribution: uniform over all partitions. Because this is simply constant, it satisfies conditions 1 and 2. Assume there are 2 particles in the system. We can thus compute $p([1,2]) = p([1],[2]) = 1/2$. When there are 3 particles we have $p([1,2,3]) = p([1,2],[3]) = p([1],[2,3]) = p([2,3],[1]) = p([1],[2],[3]) = 1/5$. Now, since $p([1,2,3]) + p([1,2],[3]) = 2/5$ is not equal to $p([1,2]) = 1/2$ we have violated consistency. A slightly more general class of distributions over partitions which satisfies both conditions 1 and 2 can be constructed by first sampling the number of clusters K from some arbitrary $p(K)$, and then sampling the partitions in that subset uniformly at random (this distribution was considered in (Consonni & Veronese, 1995) in a different context). This construction is in general not consistent[3] as the above example has shown [4].

In the above examples we maintained exchangeability over particles by using functions of invariants w.r.t. particle permutations only (e.g. count vectors $\{N_i\}$). Exchangeability over labels was achieved by making sure the distributions were symmetric w.r.t. label permutations. Alternatively, both conditions can be satisfied if we choose functions of size-ordered count vectors, $N_1 \geq N_2 \geq ... \geq N_K$. The label i in N_i in this decreasing sequence of counts should not be confused with a cluster label. For example, if during sampling the biggest cluster, say 4, with N_1 particles shrinks below the size of the runner up, say cluster 2 with N_2 particles, then they switch places and N_1 now describes cluster 2 while N_2 describes cluster 4. A general class of distributions on ordered count vectors can be described as follows:

1. Sample N_1 from some discrete distribution $P_1(N_1; D_1)$, where the domain is $D_1 = [1, .., N]$.

2. At iteration $1 < i \leq N$, compute the residual resources $R_i = R_{i-1} - N_{i-1} = N - \sum_{j=1}^{i-1} N_j$. The domain of N_i is now calculated as $D_i = [1, .., \min(R_i, N_{i-1})]$. Finally sample N_i from any discrete distribution $P_i(N_i; D_i)$ in domain D_i.

3. When $N_i = 1$ for some i, compute the new residual $R_{i+1} = R_i - 1$, and set $N_j = 1$ for $j = i+1, .., i + R_{i+1}$, where $K = i + R_{i+1}$.

The distributions $P_i(N; D)$ are completely arbitrary within their domain giving a large degree of flexibility in designing the prior. We have experimented with a Binomial distribution[5] with some fixed value for the probability of success.

Yet another way to build priors on exchangeable partitions is to write them as functions of the $\{m_j\}$ variables under the constraint that $\sum_j j m_j = N$. For instance, one could choose a distribution of the form,

$$p(Z) \propto \exp(\sum_{j=1}^{N} \alpha_j m_j)\, \mathbb{I}(\sum_{j=1}^{N} j m_j = N) \qquad (15)$$

with $\sum_{j=1}^{N} m_j = K$ and $\mathbb{I}$ the indicator function which is only 1 if the condition in its argument is satisfied, and 0 otherwise. The parameters are given by $\{\alpha_j\}$. This distribution is a special case of a "Gibbs partition" (Pitman, 2002). One could introduce interactions between m_j variables by defining a pairwise Markov random field or even more generally, a general random field model with features $\phi(\{m_j\})$. The normalization of this distribution is clearly very hard to compute, certainly given the constraint. However, all we need for Gibbs sampling is to compute the unnormalized probabilities for all possible ways to re-assign particle n, and then to normalize over these possibilities.

We can get interesting behavior for the general parameterization,

$$\alpha_j = a + b \log \Gamma(j + c) \qquad (16)$$

where c seems to have an important effect on the distribution over the (size-biased) cluster sizes. Note that for $\alpha_j = \log \alpha + \log \Gamma(j)$ we return to the DP(α) prior.

The above constructions are just two examples of many possible ways to design priors over partitions. I leave the exploration of new priors for future research. The main point I want to make is that a whole range of possibilities opens up by dropping the consistency requirement.

The efficiency of Gibbs sampling is hardly affected by the violation of consistency, provided that it is reasonably efficient to compute the (unnormalized) probability of any state $N_1, ..., N_K$. The procedure is very similar to that of a regular DP. First take particle n out of the system (cycling over all particles) and reassign it in all possible ways. Since we sample in the space of partitions, we have again the choice to join an existing cluster or to create a new cluster.

[3] There is however a choice of $p(K)$ and $p(Z|K)$ that will result in consistency. This choice can be found in (Green & Richardson, 2001) where the DP prior itself was decomposed in the above manner and hence satisfies all three conditions. A similar decomposition can presumably be found for the Pitman-Yor process.

[4] The statement in (Green & Richardson, 2001) in appendix A claiming that this distribution is consistent is therefore incorrect (private communication with P. Green).

[5] Note that we do not allow $N = 0$ so we change the Binomial slightly.

For all these possibilities compute the (possibly unnormalized) probability $p(N_1, N_2, ..)$ and *divide by the number of partitions described by the same signature*,

$$p(Z) = \frac{\prod_{i=1}^{N}(i!)^{m_i} m_i!}{N!} p(\{N_i\}) \tag{17}$$

where m_j is again the number if clusters of size j. Here we interpret Z as living in the space of partitions. For example, with 3 particles, there are 3 partitions with signature $N_1 = 2, N_2 = 1$. Finally, given these probabilities for reassignments, we sample one of these possibilities. Note that through the reassignment of particle n the order of the N_i may swap.

What did we loose when we gave up consistency? Firstly, we are no longer able to describe the process of sampling from the prior using a Chinese restaurant or other culinary metaphor, i.e. through an exchangeable prediction rule. This is perhaps less elegant, but since the Gibbs sampler does not suffer much in terms of efficiency it seems no reason to avoid inconsistent priors.

There is a second, perhaps more philosophical argument to dislike inconsistency. When we propose a prior for clustering, we probably pick our prior without considering exactly how many particles are in the dataset we need to cluster. However, imagine we start with a dataset of size N and cluster it using our inconsistent prior. Someone comes along and requests to redo the clustering but now on a subset of the original dataset of size $M < N$. If we use the same (inconsistent) prior, but now replacing N with M we are reasoning inconsistently. One can compute the probability of partitions for M particles from the distribution for N particles by removing $N - M$ particles uniformly at random. The resulting distribution will not match the distribution where we replace N with M in the original prior unless it is consistent. Hence, we are forced to defend the statement that we choose this particular prior especially for this dataset of size N. Clearly, we are free to choose any prior, but philosophically speaking it is a hard case to sell. The situation may become less unsatisfactory when we allow ourselves to learn some parameters of the prior from the data, which will now depend on N. In practice, I see no reason to avoid inconsistent priors as long as one makes sure the prior of choice makes sense for the amount of data in the dataset under consideration.

4. Pre-Learning

In this section I describe a way to learn a prior before one observes data. We'll use the general form proposed in (15). We specify our prior knowledge through a number of sufficient statistics, such as $\mathbb{E}[m_j] \;\; \forall j$ subject to $\sum_j j\mathbb{E}[m_j] = N$, and train a maximum entropy model. Maximum entropy models are attractive because they assume as little as possible about the other properties of the distribution but do satisfy the marginal constraints. Our task is to learn the parameters of the model α_j which take the role of Lagrange multipliers in the maximum entropy setting. Note that the constraint $\sum_j jm_j = N$ couples all m_j variables together, so it is unlikely that we can express α_j as an analytic expression of $\{m_j\}$.

We have adopted a very simple yet effective approach to obtain values for the parameters. We simply run the Gibbs sampler and interrupt it periodically to update the parameters according to

$$\alpha_j \leftarrow \alpha_j + \eta(\bar{m}_j - \mathbb{E}[m_j]) \tag{18}$$

where η is a step-size, $\bar{m}_j$ is the prior expected value for this statistic (i.e. it is supplied by the modeler) and $\mathbb{E}[m_j]$ is the average of m_j collected from the Gibbs sampler. This approach is sound as long as the change in the distribution through these parameter updates is slow compared to the convergence time of the Gibbs sampler.

In figure (4) we show a few histograms of average m_j values that were used to train a maximum entropy model together with the histogram obtained from a sampler with the learned model. We note that for a uniform distribution (third column) the variance of m_j must necessarily be very high because the probability of a single cluster containing all objects must be as high as the probability of N separate clusters. This seems to have affected the results.

5. Discussion

We have discussed methods to design priors for the problem of clustering N particles. The first method is based on relaxing the requirement that the prior should be exchangeable over cluster relabeling. This indeed results in a much more general class of priors but with the disadvantage that the Gibbs sampler operates in the space of explicit cluster labels and not in the much to be preferred space of partitions. It has gone unnoticed in the literature that this may in fact result in a mild clustering bias. The second method is based on relaxing the requirement of consistency. In this case we seem to loose the ability to describe probabilities using a type of Chinese restaurant metaphor. This is unfortunate, but in no way restrictive for running an efficient Gibbs sampler. The upside is that we have gained enormous flexibility in designing our priors. We have described one method to generate priors over partitions, but we feel many more are possible. In particular we are studying the usefulness of Gibbs partitions in this respect.

The Gibbs sampler proposed in this paper is only one possible way to sample from the posterior. In particular it seems straightforward to include cluster merge and split moves into the sampler, resulting in a hybrid Gibbs-reversible jump sampler.

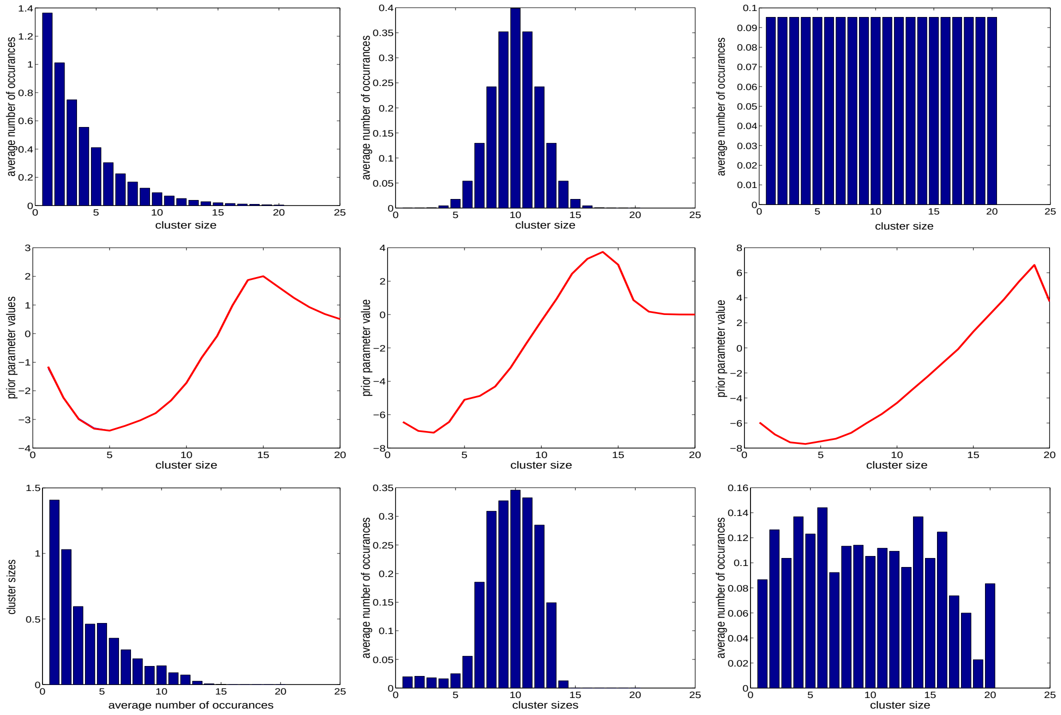

Figure 3. Top row: 3 examples of input distributions over m_j variables ($N = 20$.) Middle row: Learned parameter values α_j (learning rate $\eta = 0.1$, 500 parameter updates, 10 samples per update, 100 iterations burn-in.) Bottom row: average histogram for m_j variables from a Gibbs sampler with learned α_j variables (average over 1000 samples sub-sampled from $10,000$ Gibbs samples, first 100 samples discarded.)

We like to emphasize that placing a prior directly on partitions is as much an "infinite model" as a model that was obtained by explicitly taking the limit of finite distribution to infinity. There is an interesting analogy in kernel methods where one can take the limit of the number of features to infinity or directly work with kernel matrices of particles.

By the same reasoning we can also extend these ideas to processes on other combinatorial structures such as the "Indian Buffet Process" on binary matrices (Griffiths & Ghahramani, 2006). We have been able to construct flexible priors which are exchangeable over both particles and feature labels by requiring them to be functions of invariants such as (i) the number of particles with exactly j features and (ii) the number of features with exactly i particles. Moreover, reasonably efficient MCMC samplers can be constructed for these more general distributions as well.

In conclusion, I like to challenge the view that priors for infinite mixture models should be either based on DP or Pitman-Yor processes. By relaxing some conditions we can gain flexibility in the design of our priors.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0535278 and IIS-0431085.

References

Blackwell, D., & MacQueen, J. (1973). Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, *1*, 353–285.

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

Bush, C., & MacEachern, S. (1996). A semiparametric bayesian model for randomised block designs. *Biometrika*, *83*, 275–285.

Consonni, G., & Veronese, P. (1995). A Bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, *90(431)*, 935–944.

Escobar, M., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, *90*, 577–588.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*, 209–230.

Green, P., & Richardson, S. (2001). Modelling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics*, *28*, 355377.

Griffiths, T., & Ghahramani, Z. (2006). Infinite latent feature models and the indian buffet process. *Advances in Neural Information Processing Systems 18* (pp. 475–482).

Ishwaran, H., & James, L. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, *96*, 161–173.

Ishwaran, H., & James, L. (2003). Some further developments for stick-breaking priors: finite and infinite clustering and classification. *Sankhya Series A*, *65*, 577–592.

MacEachern, S., & Müller, P. (1998). Estimating mixture of Dirichlet process models. *Communications in Statistics*, *7*, 223–238.

Neal, R. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *9*, 283–297.

Pitman, J. (2002). *Combinatorial stochastic processes* (Technical Report). Dept. Statistics, U.C. Berkeley. Lecture notes for St. Flour course, Technical Report no.621.

Porteous, I., Ihler, A., Smyth, P., & Welling, M. (2006). Gibbs sampling for (coupled) infinite mixture models in the stick-breaking representation. *Proc. of the Conf. on Uncertainty in Artificial Intelligence*. Pittsburgh, PA.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*, 639–650.