Journal of Machine Learning Research 12 (2011) 1185-1224 Submitted 3/10; Revised 3/11; Published 4/11

The Indian Buffet Process: An Introduction and Review

Thomas L. Griffiths TOM_GRIFFITHS@BERKELEY.EDU
Department of Psychology
University of California, Berkeley
Berkeley, CA 94720-1650, USA

Zoubin Ghahramani[*] ZOUBIN@ENG.CAM.AC.UK
Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK

Editor: David M. Blei

Abstract

The Indian buffet process is a stochastic process defining a probability distribution over equivalence classes of sparse binary matrices with a finite number of rows and an unbounded number of columns. This distribution is suitable for use as a prior in probabilistic models that represent objects using a potentially infinite array of features, or that involve bipartite graphs in which the size of at least one class of nodes is unknown. We give a detailed derivation of this distribution, and illustrate its use as a prior in an infinite latent feature model. We then review recent applications of the Indian buffet process in machine learning, discuss its extensions, and summarize its connections to other stochastic processes.

Keywords: nonparametric Bayes, Markov chain Monte Carlo, latent variable models, Chinese restaurant processes, beta process, exchangeable distributions, sparse binary matrices

1. Introduction

Unsupervised learning aims to recover the latent structure responsible for generating observed data. One of the key problems faced by unsupervised learning algorithms is thus determining the amount of latent structure—the number of clusters, dimensions, or variables—needed to account for the regularities expressed in the data. Often, this is treated as a model selection problem, choosing the model with the dimensionality that results in the best performance. This treatment of the problem assumes that there is a single, finite-dimensional representation that correctly characterizes the properties of the observed objects. An alternative is to assume that the amount of latent structure is actually potentially unbounded, and that the observed objects only manifest a sparse subset of those classes or features (Rasmussen and Ghahramani, 2001).

The assumption that the observed data manifest a subset of an unbounded amount of latent structure is often used in nonparametric Bayesian statistics, and has recently become increasingly popular in machine learning. In particular, this assumption is made in Dirichlet process mixture models, which are used for nonparametric density estimation (Antoniak, 1974; Escobar and West, 1995; Ferguson, 1983; Neal, 2000). Under one interpretation of a Dirichlet process mixture model, each datapoint is assigned to a latent class, and each class is associated with a distribution over

[*]. Also at the Machine Learning Department, Carnegie Mellon University, Pittsburgh PA 15213, USA.

observable properties. The prior distribution over assignments of datapoints to classes is specified in such a way that the number of classes used by the model is bounded only by the number of objects, making Dirichlet process mixture models "infinite" mixture models (Rasmussen, 2000).

Recent work has extended Dirichlet process mixture models in a number of directions, making it possible to use nonparametric Bayesian methods to discover the kinds of structure common in machine learning: hierarchies (Blei et al., 2004; Heller and Ghahramani, 2005; Neal, 2003; Teh et al., 2008), topics and syntactic classes (Teh et al., 2004) and the objects appearing in images (Sudderth et al., 2006). However, the fact that all of these models are based upon the Dirichlet process limits the kinds of latent structure that they can express. In many of these models, each object described in a data set is associated with a latent variable that picks out a single class or parameter responsible for generating that datapoint. In contrast, many models used in unsupervised learning represent each object as having multiple features or being produced by multiple causes. For instance, we could choose to represent each object with a binary vector, with entries indicating the presence or absence of each feature (e.g., Ueda and Saito, 2003), allow each feature to take on a continuous value, representing datapoints with locations in a latent space (e.g., Jolliffe, 1986), or define a factorial model, in which each feature takes on one of a discrete set of values (e.g., Zemel and Hinton, 1994; Ghahramani, 1995). Infinite versions of these models are difficult to define using the Dirichlet process.

In this paper, we summarize recent work exploring the extension of this nonparametric approach to models in which objects are represented using an unknown number of latent features. Following Griffiths and Ghahramani (2005, 2006), we provide a detailed derivation of a distribution that can be used to define probabilistic models that represent objects with infinitely many binary features, and can be combined with priors on feature values to produce factorial and continuous representations. This distribution can be specified in terms of a simple stochastic process called the *Indian buffet process*, by analogy to the *Chinese restaurant process* used in Dirichlet process mixture models. We illustrate how the Indian buffet process can be used to specify prior distributions in latent feature models, using a simple linear-Gaussian model to show how such models can be defined and used.

The Indian buffet process can also be used to define a prior distribution in any setting where the latent structure expressed in data can be expressed in the form of a binary matrix with a finite number of rows and infinite number of columns, such as the adjacency matrix of a bipartite graph where one class of nodes is of unknown size, or the adjacency matrix for a Markov process with an unbounded set of states. As a consequence, this approach has found a number of recent applications within machine learning. We review these applications, summarizing some of the innovations that have been introduced in order to use the Indian buffet process in different settings, as well as extensions to the basic model and alternative inference algorithms. We also describe some of the interesting connections to other stochastic processes that have been identified. As for the Chinese restaurant process, we can arrive at the Indian buffet process in a number of different ways: as the infinite limit of a finite model, via the constructive specification of an infinite model, or by marginalizing out an underlying measure. Each perspective provides different intuitions, and suggests different avenues for designing inference algorithms and generalizations.

The plan of the paper is as follows. Section 2 summarizes the principles behind infinite mixture models, focusing on the prior on class assignments assumed in these models, which can be defined in terms of a simple stochastic process—the Chinese restaurant process. We then develop a distribution on infinite binary matrices by considering how this approach can be extended to the case where objects are represented with multiple binary features. Section 3 discusses the role of a such a

distribution in defining infinite latent feature models. Section 4 derives the distribution, making use of the Indian buffet process. Section 5 illustrates how this distribution can be used as a prior in a nonparametric Bayesian model, defining an infinite-dimensional linear-Gaussian model, deriving a sampling algorithm for inference in this model, and applying it to two simple data sets. Section 6 describes further applications of this approach, both in latent feature models and for inferring graph structures, and Section 7 discusses recent work extending the Indian buffet process and providing connections to other stochastic processes. Section 8 presents conclusions and directions for future work.

2. Latent Class Models

Assume we have N objects, with the ith object having D observable properties represented by a row vector $\mathbf{x}_i$. In a latent class model, such as a mixture model, each object is assumed to belong to a single class, c_i, and the properties $\mathbf{x}_i$ are generated from a distribution determined by that class. Using the matrix $\mathbf{X} = \left[\mathbf{x}_1^T\ \mathbf{x}_2^T\ \cdots\ \mathbf{x}_N^T\right]^T$ to indicate the properties of all N objects, and the vector $\mathbf{c} = [c_1\ c_2\ \cdots\ c_N]^T$ to indicate their class assignments, the model is specified by a prior over assignment vectors $P(\mathbf{c})$, and a distribution over property matrices conditioned on those assignments, $p(\mathbf{X}|\mathbf{c})$.[1] These two distributions can be dealt with separately: $P(\mathbf{c})$ specifies the number of classes and their relative probability, while $p(\mathbf{X}|\mathbf{c})$ determines how these classes relate to the properties of objects. In this section, we will focus on the prior over assignment vectors, $P(\mathbf{c})$, showing how such a prior can be defined without placing an upper bound on the number of classes.

2.1 Finite Mixture Models

Mixture models assume that the assignment of an object to a class is independent of the assignments of all other objects. If there are K classes, we have

$$P(\mathbf{c}|\theta) = \prod_{i=1}^{N} P(c_i|\theta) = \prod_{i=1}^{N} \theta_{c_i},$$

where θ is a multinomial distribution over those classes, and θ_k is the probability of class k under that distribution. Under this assumption, the probability of the properties of all N objects $\mathbf{X}$ can be written as

$$p(\mathbf{X}|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(\mathbf{x}_i|c_i = k)\,\theta_k. \tag{1}$$

The distribution from which each $\mathbf{x}_i$ is generated is thus a *mixture* of the K class distributions $p(\mathbf{x}_i|c_i = k)$, with θ_k determining the weight of class k.

The mixture weights θ can be treated as a parameter to be estimated. In Bayesian approaches to mixture modeling, θ is assumed to follow a prior distribution $p(\theta)$, with a standard choice being a symmetric Dirichlet distribution. The Dirichlet distribution on multinomials over K classes has parameters $\alpha_1, \alpha_2, \ldots, \alpha_K$, and is conjugate to the multinomial (e.g., Bernardo and Smith, 1994).

1. We will use $P(\cdot)$ to indicate probability mass functions, and $p(\cdot)$ to indicate probability density functions. We will assume that $\mathbf{x}_i \in \mathbb{R}^D$, and $p(\mathbf{X}|\mathbf{c})$ is thus a density, although variants of the models we discuss also exist for discrete data.

The probability density for the parameter θ of a multinomial distribution is given by

$$p(\theta) = \frac{\prod_{k=1}^{K} \theta_k^{\alpha_k - 1}}{D(\alpha_1, \alpha_2, \ldots, \alpha_K)},$$

in which $D(\alpha_1, \alpha_2, \ldots, \alpha_K)$ is the Dirichlet normalizing constant

$$\begin{aligned} D(\alpha_1, \alpha_2, \ldots, \alpha_K) &= \int_{\Delta_K} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} d\theta \\ &= \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}, \end{aligned} \tag{2}$$

where Δ_K is the simplex of multinomials over K classes, and $\Gamma(\cdot)$ is the gamma, or generalized factorial, function, with $\Gamma(m) = (m-1)!$ for any non-negative integer m. In a *symmetric* Dirichlet distribution, all α_k are equal. For example, we could take $\alpha_k = \frac{\alpha}{K}$ for all k. In this case, Equation 2 becomes

$$D(\tfrac{\alpha}{K}, \tfrac{\alpha}{K}, \ldots, \tfrac{\alpha}{K}) = \frac{\Gamma(\frac{\alpha}{K})^K}{\Gamma(\alpha)},$$

and the mean of θ is the multinomial that is uniform over all classes.

The probability model that we have defined is

$$\begin{aligned} \theta \,|\, \alpha &\sim \text{Dirichlet}(\tfrac{\alpha}{K}, \tfrac{\alpha}{K}, \ldots, \tfrac{\alpha}{K}), \\ c_i \,|\, \theta &\sim \text{Discrete}(\theta) \end{aligned}$$

where $\text{Discrete}(\theta)$ is the multiple-outcome analogue of a Bernoulli event, where the probabilities of the outcomes are specified by θ (i.e., $P(c_i = k|\theta) = \theta_k$). The dependencies among variables in this model are shown in Figure 1. Having defined a prior on θ, we can simplify this model by integrating over all values of θ rather than representing them explicitly. The marginal probability of an assignment vector $\mathbf{c}$, integrating over all values of θ, is

$$\begin{aligned} P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{n} P(c_i|\theta)\; p(\theta)\, d\theta \\ &= \int_{\Delta_K} \frac{\prod_{k=1}^{K} \theta_k^{m_k + \alpha/K - 1}}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \ldots, \frac{\alpha}{K})}\, d\theta \\ &= \frac{D(m_1 + \frac{\alpha}{K}, m_2 + \frac{\alpha}{K}, \ldots, m_k + \frac{\alpha}{K})}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \ldots, \frac{\alpha}{K})} \\ &= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)}, \end{aligned} \tag{3}$$

where $m_k = \sum_{i=1}^{N} \delta(c_i = k)$ is the number of objects assigned to class k. The tractability of this integral is a result of the fact that the Dirichlet is conjugate to the multinomial.

Equation 3 defines a joint probability distribution for all class assignments $\mathbf{c}$ in which individual class assignments are not independent. Rather, they are *exchangeable* (Bernardo and Smith, 1994), with the probability of an assignment vector remaining the same when the indices of the objects are permuted. Exchangeability is a desirable property in a distribution over class assignments, because

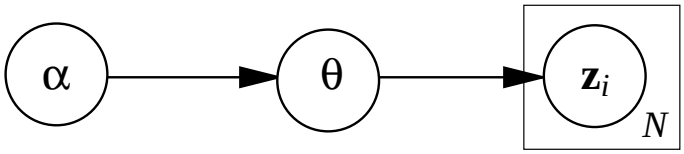

Figure 1: Graphical model for the Dirichlet-multinomial model used in defining the Chinese restaurant process. Nodes are variables, arrows indicate dependencies, and plates (Buntine, 1994) indicate replicated structures.

we have no special knowledge about the objects that would justify treating them differently from one another. However, the distribution on assignment vectors defined by Equation 3 assumes an upper bound on the number of classes of objects, since it only allows assignments of objects to up to K classes.

2.2 Infinite Mixture Models

Intuitively, defining an infinite mixture model means that we want to specify the probability of $\mathbf{X}$ in terms of infinitely many classes, modifying Equation 1 to become

$$p(\mathbf{X}|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i|c_i = k)\,\theta_k,$$

where θ is an infinite-dimensional multinomial distribution. In order to repeat the argument above, we would need to define a prior, $p(\theta)$, on infinite-dimensional multinomials, and compute the probability of $\mathbf{c}$ by integrating over θ. This is essentially the strategy that is taken in deriving infinite mixture models from the Dirichlet process (Antoniak, 1974; Ferguson, 1983; Ishwaran and James, 2001; Sethuraman, 1994). Instead, we will work directly with the distribution over assignment vectors given in Equation 3, considering its limit as the number of classes approaches infinity (cf., Green and Richardson, 2001; Neal, 1992, 2000).

Expanding the gamma functions in Equation 3 using the recursion $\Gamma(x) = (x-1)\Gamma(x-1)$ and cancelling terms produces the following expression for the probability of an assignment vector $\mathbf{c}$:

$$P(\mathbf{c}) = \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{j=1}^{m_k-1} (j + \tfrac{\alpha}{K})\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \tag{4}$$

where K_+ is the number of classes for which $m_k > 0$, and we have re-ordered the indices such that $m_k > 0$ for all $k \leq K_+$. There are K^N possible values for $\mathbf{c}$, which diverges as $K \to \infty$. As this happens, the probability of any single set of class assignments goes to 0. Since $K_+ \leq N$ and N is finite, it is clear that $P(\mathbf{c}) \to 0$ as $K \to \infty$, since $\frac{1}{K} \to 0$. Consequently, we will define a distribution over equivalence classes of assignment vectors, rather than the vectors themselves.

Specifically, we will define a distribution on *partitions* of objects. In our setting, a partition is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter. Two assignment vectors that result in the same division of objects correspond to the same partition. For example, if we had three objects, the class

assignments $\{c_1, c_2, c_3\} = \{1, 1, 2\}$ would correspond to the same partition as $\{2, 2, 1\}$, since all that differs between these two cases is the labels of the classes. A partition thus defines an equivalence class of assignment vectors, which we denote $[\mathbf{c}]$, with two assignment vectors belonging to the same equivalence class if they correspond to the same partition. A distribution over partitions is sufficient to allow us to define an infinite mixture model, provided the prior distribution on the parameters is the same for all classes. In this case, these equivalence classes of class assignments are the same as those induced by identifiability: $p(\mathbf{X}|\mathbf{c})$ is the same for all assignment vectors $\mathbf{c}$ that correspond to the same partition, so we can apply statistical inference at the level of partitions rather than the level of assignment vectors.

Assume we have a partition of N objects into K_+ subsets, and we have $K = K_0 + K_+$ class labels that can be applied to those subsets. Then there are $\frac{K!}{K_0!}$ assignment vectors $\mathbf{c}$ that belong to the equivalence class defined by that partition, $[\mathbf{c}]$. We can define a probability distribution over partitions by summing over all class assignments that belong to the equivalence class defined by each partition. The probability of each of those class assignments is equal under the distribution specified by Equation 4, so we obtain

$$\begin{aligned} P([\mathbf{c}]) &= \sum_{\mathbf{c} \in [\mathbf{c}]} P(\mathbf{c}) \\ &= \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{j=1}^{m_k - 1} (j + \tfrac{\alpha}{K}) \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}. \end{aligned}$$

Rearranging the first two terms, we can compute the limit of the probability of a partition as $K \to \infty$, which is

$$\begin{aligned} &\lim_{K \to \infty} \alpha^{K_+} \cdot \frac{K!}{K_0! K^{K_+}} \cdot \left(\prod_{k=1}^{K_+} \prod_{j=1}^{m_k - 1} (j + \tfrac{\alpha}{K}) \right) \cdot \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \\ &= \alpha^{K_+} \cdot \quad 1 \quad \cdot \left(\prod_{k=1}^{K_+} (m_k - 1)! \right) \cdot \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}. \end{aligned} \tag{5}$$

The details of the steps taken in computing this limit are given in Appendix A. These limiting probabilities define a valid distribution over partitions, and thus over equivalence classes of class assignments, providing a prior over class assignments for an infinite mixture model. Objects are exchangeable under this distribution, just as in the finite case: the probability of a partition is not affected by the ordering of the objects, since it depends only on the counts m_k.

As noted above, the distribution over partitions specified by Equation 5 can be derived in a variety of ways—by taking limits (Green and Richardson, 2001; Neal, 1992, 2000), from the Dirichlet process (Blackwell and MacQueen, 1973), or from other equivalent stochastic processes (Ishwaran and James, 2001; Sethuraman, 1994). We will briefly discuss a simple process that produces the same distribution over partitions: the Chinese restaurant process.

2.3 The Chinese Restaurant Process

The Chinese restaurant process (CRP) was named by Jim Pitman and Lester Dubins, based upon a metaphor in which the objects are customers in a restaurant, and the classes are the tables at which they sit (the process first appears in Aldous 1985, where it is attributed to Pitman, although

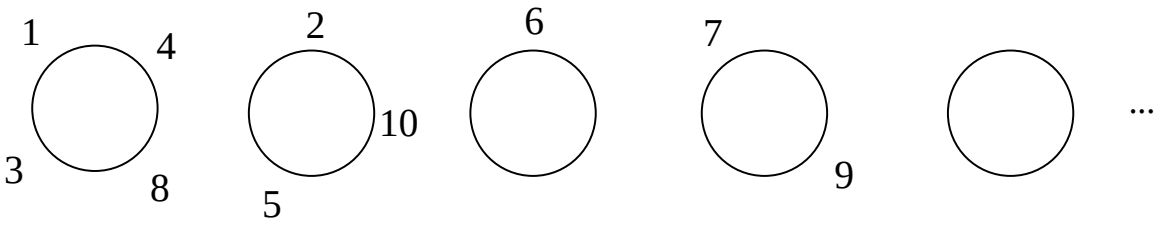

Figure 2: A partition induced by the Chinese restaurant process. Numbers indicate customers (objects), circles indicate tables (classes).

it is identical to the extended Polya urn scheme introduced by Blackwell and MacQueen 1973). Imagine a restaurant with an infinite number of tables, each with an infinite number of seats.[2] The customers enter the restaurant one after another, and each choose a table at random. In the CRP with parameter α, each customer chooses an occupied table with probability proportional to the number of occupants, and chooses the next vacant table with probability proportional to α. For example, Figure 2 shows the state of a restaurant after 10 customers have chosen tables using this procedure. The first customer chooses the first table with probability $\frac{\alpha}{\alpha} = 1$. The second customer chooses the first table with probability $\frac{1}{1+\alpha}$, and the second table with probability $\frac{\alpha}{1+\alpha}$. After the second customer chooses the second table, the third customer chooses the first table with probability $\frac{1}{2+\alpha}$, the second table with probability $\frac{1}{2+\alpha}$, and the third table with probability $\frac{\alpha}{2+\alpha}$. This process continues until all customers have seats, defining a distribution over allocations of people to tables, and, more generally, objects to classes. Extensions of the CRP and connections to other stochastic processes are pursued in depth by Pitman (2002).

The distribution over partitions induced by the CRP is the same as that given in Equation 5. If we assume an ordering on our N objects, then we can assign them to classes sequentially using the method specified by the CRP, letting objects play the role of customers and classes play the role of tables. The ith object would be assigned to the kth class with probability

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

where m_k is the number of objects currently assigned to class k, and K_+ is the number of classes for which $m_k > 0$. If all N objects are assigned to classes via this process, the probability of a partition of objects $\mathbf{c}$ is that given in Equation 5. The CRP thus provides an intuitive means of specifying a prior for infinite mixture models, as well as revealing that there is a simple sequential process by which exchangeable class assignments can be generated.

2.4 Inference by Gibbs Sampling

Inference in an infinite mixture model is only slightly more complicated than inference in a mixture model with a finite, fixed number of classes. The standard algorithm used for inference in infinite mixture models is Gibbs sampling (Bush and MacEachern, 1996; Neal, 2000). Gibbs sampling

2. Pitman and Dubins, both statisticians at the University of California, Berkeley, were inspired by the apparently infinite capacity of Chinese restaurants in San Francisco when they named the process.

is a Markov chain Monte Carlo (MCMC) method, in which variables are successively sampled from their distributions when conditioned on the current values of all other variables (Geman and Geman, 1984). This process defines a Markov chain, which ultimately converges to the distribution of interest (see Gilks et al., 1996). Recent work has also explored variational inference algorithms for these models (Blei and Jordan, 2006), a topic we will return to later in the paper.

Implementing a Gibbs sampler requires deriving the full conditional distribution for all variables to be sampled. In a mixture model, these variables are the class assignments $\mathbf{c}$. The relevant full conditional distribution is $P(c_i|\mathbf{c}_{-i},\mathbf{X})$, the probability distribution over c_i conditioned on the class assignments of all other objects, $\mathbf{c}_{-i}$, and the data, $\mathbf{X}$. By applying Bayes' rule, this distribution can be expressed as

$$P(c_i = k|\mathbf{c}_{-i},\mathbf{X}) \propto p(\mathbf{X}|\mathbf{c})P(c_i = k|\mathbf{c}_{-i}),$$

where only the second term on the right hand side depends upon the distribution over class assignments, $P(\mathbf{c})$. Here we assume that the parameters associated with each class can be integrated out, so we that the probability of the data depends only on the class assignment. This is possible when a conjugate prior is used on these parameters. For details, and alternative algorithms that can be used when this assumption is violated, see Neal (2000).

In a finite mixture model with $P(\mathbf{c})$ defined as in Equation 3, we can compute $P(c_i = k|\mathbf{c}_{-i})$ by integrating over θ, obtaining

$$\begin{aligned} P(c_i = k|\mathbf{c}_{-i}) &= \int P(c_i = k|\theta)p(\theta|\mathbf{c}_{-i})\,d\theta \\ &= \frac{m_{-i,k} + \frac{\alpha}{K}}{N-1+\alpha}, \end{aligned} \tag{6}$$

where $m_{-i,k}$ is the number of objects assigned to class k, not including object i. This is the posterior predictive distribution for a multinomial distribution with a Dirichlet prior.

In an infinite mixture model with a distribution over class assignments defined as in Equation 5, we can use exchangeability to find the full conditional distribution. Since it is exchangeable, $P([\mathbf{c}])$ is unaffected by the ordering of objects. Thus, we can choose an ordering in which the ith object is the last to be assigned to a class. It follows directly from the definition of the Chinese restaurant process that

$$P(c_i = k|\mathbf{c}_{-i}) = \begin{cases} \frac{m_{-i,k}}{N-1+\alpha} & m_{-i,k} > 0 \\ \frac{\alpha}{N-1+\alpha} & k = K_{-i,+}+1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $K_{-i,+}$ is the number of classes for which $m_{-i,k} > 0$. The same result can be found by taking the limit of the full conditional distribution in the finite model, given by Equation 6 (Neal, 2000).

When combined with some choice of $p(\mathbf{X}|\mathbf{c})$, Equations 6 and 7 are sufficient to define Gibbs samplers for finite and infinite mixture models respectively. Demonstrations of Gibbs sampling in infinite mixture models are provided by Neal (2000) and Rasmussen (2000). Similar MCMC algorithms are presented in Bush and MacEachern (1996), West et al. (1994), Escobar and West (1995) and Ishwaran and James (2001). Algorithms that go beyond the local changes in class assignments allowed by a Gibbs sampler are given by Jain and Neal (2004) and Dahl (2003).

2.5 Summary

Our review of infinite mixture models serves three purposes: it shows that infinite statistical models can be defined by specifying priors over infinite combinatorial objects; it illustrates how these priors

can be derived by taking the limit of priors for finite models; and it demonstrates that inference in these models can remain possible, despite the large hypothesis spaces they imply. However, infinite mixture models are still fundamentally limited in their representation of objects, assuming that each object can only belong to a single class. In the next two sections, we use the insights underlying infinite mixture models to derive methods for representing objects in terms of infinitely many latent features, leading us to derive a distribution on infinite binary matrices.

3. Latent Feature Models

In a latent feature model, each object is represented by a vector of latent feature values $\mathbf{f}_i$, and the properties $\mathbf{x}_i$ are generated from a distribution determined by those latent feature values. Latent feature values can be continuous, as in factor analysis (Roweis and Ghahramani, 1999) and probabilistic principal component analysis (PCA; Tipping and Bishop, 1999), or discrete, as in cooperative vector quantization (CVQ; Zemel and Hinton, 1994; Ghahramani, 1995). In the remainder of this section, we will assume that feature values are continuous. Using the matrix $\mathbf{F} = \left[\mathbf{f}_1^T \mathbf{f}_2^T \cdots \mathbf{f}_N^T\right]^T$ to indicate the latent feature values for all N objects, the model is specified by a prior over features, $p(\mathbf{F})$, and a distribution over observed property matrices conditioned on those features, $p(\mathbf{X}|\mathbf{F})$. As with latent class models, these distributions can be dealt with separately: $p(\mathbf{F})$ specifies the number of features, their probability, and the distribution over values associated with each feature, while $p(\mathbf{X}|\mathbf{F})$ determines how these features relate to the properties of objects. Our focus will be on $p(\mathbf{F})$, showing how such a prior can be defined without placing an upper bound on the number of features.

We can break the matrix $\mathbf{F}$ into two components: a binary matrix $\mathbf{Z}$ indicating which features are possessed by each object, with $z_{ik} = 1$ if object i has feature k and 0 otherwise, and a second matrix $\mathbf{V}$ indicating the value of each feature for each object. $\mathbf{F}$ can be expressed as the elementwise (Hadamard) product of $\mathbf{Z}$ and $\mathbf{V}$, $\mathbf{F} = \mathbf{Z} \otimes \mathbf{V}$, as illustrated in Figure 3. In many latent feature models, such as PCA and CVQ, objects have non-zero values on every feature, and every entry of $\mathbf{Z}$ is 1. In *sparse* latent feature models (e.g., sparse PCA; d'Aspremont et al., 2004; Jolliffe and Uddin, 2003; Zou et al., 2006) only a subset of features take on non-zero values for each object, and $\mathbf{Z}$ picks out these subsets.

A prior on $\mathbf{F}$ can be defined by specifying priors for $\mathbf{Z}$ and $\mathbf{V}$ separately, with $p(\mathbf{F}) = P(\mathbf{Z})p(\mathbf{V})$. We will focus on defining a prior on $\mathbf{Z}$, since the effective dimensionality of a latent feature model is determined by $\mathbf{Z}$. Assuming that $\mathbf{Z}$ is sparse, we can define a prior for infinite latent feature models by defining a distribution over infinite binary matrices. Our analysis of latent class models provides two desiderata for such a distribution: objects should be exchangeable, and inference should be tractable. It also suggests a method by which these desiderata can be satisfied: start with a model that assumes a finite number of features, and consider the limit as the number of features approaches infinity.

4. A Distribution on Infinite Sparse Binary Matrices

In this section, we derive a distribution on infinite binary matrices by starting with a simple model that assumes K features, and then taking the limit as $K \to \infty$. The resulting distribution corresponds to a simple generative process, which we term the Indian buffet process.

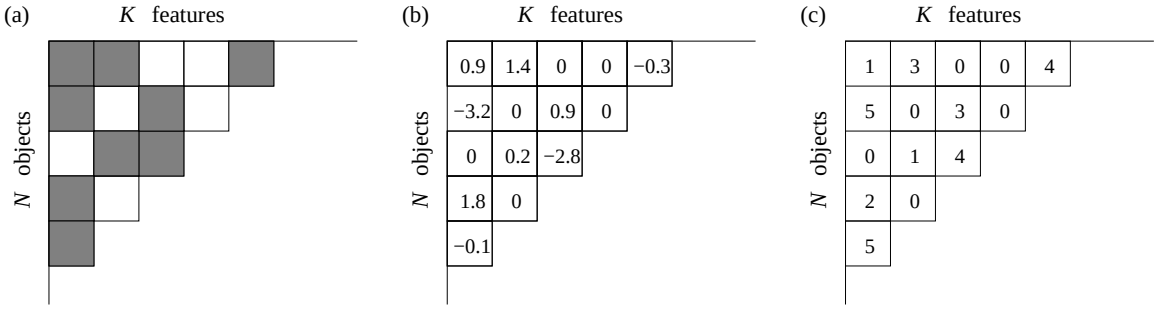

Figure 3: Feature matrices. A binary matrix $\mathbf{Z}$, as shown in (a), can be used as the basis for sparse infinite latent feature models, indicating which features take non-zero values. Element-wise multiplication of $\mathbf{Z}$ by a matrix $\mathbf{V}$ of continuous values gives a representation like that shown in (b). If $\mathbf{V}$ contains discrete values, we obtain a representation like that shown in (c).

4.1 A Finite Feature Model

We have N objects and K features, and the possession of feature k by object i is indicated by a binary variable z_{ik}. Each object can possess multiple features. The z_{ik} thus form a binary $N \times K$ feature matrix, $\mathbf{Z}$. We will assume that each object possesses feature k with probability π_k, and that the features are generated independently. In contrast to the class models discussed above, for which $\sum_k \theta_k = 1$, the probabilities π_k can each take on any value in $[0,1]$. Under this model, the probability of a matrix $\mathbf{Z}$ given $\pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$, is

$$P(\mathbf{Z}|\pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} P(z_{ik}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1-\pi_k)^{N-m_k},$$

where $m_k = \sum_{i=1}^{N} z_{ik}$ is the number of objects possessing feature k.

We can define a prior on π by assuming that each π_k follows a beta distribution. The beta distribution has parameters r and s, and is conjugate to the binomial. The probability of any π_k under the Beta(r,s) distribution is given by

$$p(\pi_k) = \frac{\pi_k^{r-1}(1-\pi_k)^{s-1}}{B(r,s)},$$

where $B(r,s)$ is the beta function,

$$\begin{aligned} B(r,s) &= \int_0^1 \pi_k^{r-1}(1-\pi_k)^{s-1}\, d\pi_k \\ &= \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)}. \end{aligned} \tag{8}$$

We will take $r = \frac{\alpha}{K}$ and $s = 1$, so Equation 8 becomes

$$B(\tfrac{\alpha}{K}, 1) = \frac{\Gamma(\frac{\alpha}{K})}{\Gamma(1+\frac{\alpha}{K})} = \frac{K}{\alpha},$$

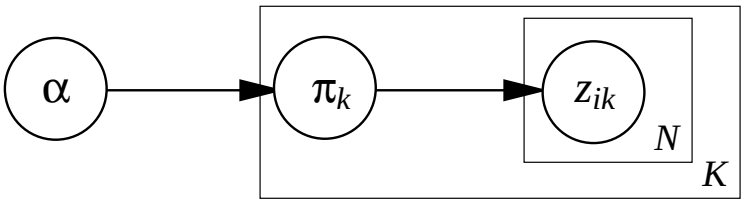

Figure 4: Graphical model for the beta-binomial model used in defining the Indian buffet process. Nodes are variables, arrows indicate dependencies, and plates (Buntine, 1994) indicate replicated structures.

exploiting the recursive definition of the gamma function.[3]

The probability model we have defined is

$$\begin{aligned} \pi_k \,|\, \alpha &\sim \text{Beta}(\tfrac{\alpha}{K}, 1), \\ z_{ik} \,|\, \pi_k &\sim \text{Bernoulli}(\pi_k). \end{aligned} \tag{9}$$

Each z_{ik} is independent of all other assignments, conditioned on π_k, and the π_k are generated independently. A graphical model illustrating the dependencies among these variables is shown in Figure 4. Having defined a prior on π, we can simplify this model by integrating over all values for π rather than representing them explicitly. The marginal probability of a binary matrix $\mathbf{Z}$ is

$$\begin{aligned} P(\mathbf{Z}) &= \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik}|\pi_k) \right) p(\pi_k)\, d\pi_k \\ &= \prod_{k=1}^{K} \frac{B(m_k + \frac{\alpha}{K}, N - m_k + 1)}{B(\frac{\alpha}{K}, 1)} \\ &= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}. \end{aligned} \tag{10}$$

Again, the result follows from conjugacy, this time between the binomial and beta distributions. This distribution is exchangeable, depending only on the counts m_k.

This model has the important property that the expectation of the number of non-zero entries in the matrix $\mathbf{Z}$, $E\left[\mathbf{1}^T\mathbf{Z}\mathbf{1}\right] = E\left[\sum_{ik} z_{ik}\right]$, has an upper bound that is independent of K. Since each column of $\mathbf{Z}$ is independent, the expectation is K times the expectation of the sum of a single column, $E\left[\mathbf{1}^T\mathbf{z}_k\right]$. This expectation is easily computed,

$$E\left[\mathbf{1}^T\mathbf{z}_k\right] = \sum_{i=1}^{N} E(z_{ik}) = \sum_{i=1}^{N} \int_0^1 \pi_k p(\pi_k)\, d\pi_k = N\frac{\frac{\alpha}{K}}{1 + \frac{\alpha}{K}}, \tag{11}$$

where the result follows from the fact that the expectation of a Beta(r,s) random variable is $\frac{r}{r+s}$. Consequently, $E\left[\mathbf{1}^T\mathbf{Z}\mathbf{1}\right] = KE\left[\mathbf{1}^T\mathbf{z}_k\right] = \frac{N\alpha}{1+\frac{\alpha}{K}}$. For finite K, the expectation of the number of entries in $\mathbf{Z}$ is bounded above by $N\alpha$.

3. The motivation for choosing $r = \frac{\alpha}{K}$ will be clear when we take the limit $K \to \infty$ in Section 4.3, while the choice of $s = 1$ will be relaxed in Section 7.1.

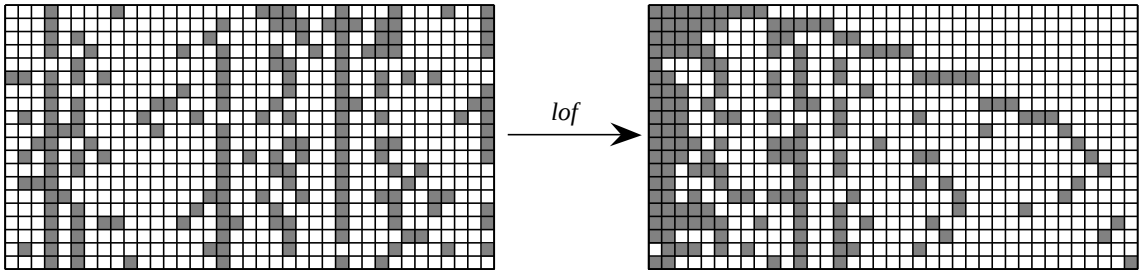

Figure 5: Binary matrices and the left-ordered form. The binary matrix on the left is transformed into the left-ordered binary matrix on the right by the function $lof(\cdot)$. This left-ordered matrix was generated from the exchangeable Indian buffet process with $\alpha = 10$. Empty columns are omitted from both matrices.

4.2 Equivalence Classes

In order to find the limit of the distribution specified by Equation 10 as $K \to \infty$, we need to define equivalence classes of binary matrices—the analogue of partitions for assignment vectors. Identifying these equivalence classes makes it easier to be precise about the objects over which we are defining probability distributions, but the reader who is satisfied with the intuitive idea of taking the limit as $K \to \infty$ can safely skip the technical details presented in this section.

Our equivalence classes will be defined with respect to a function on binary matrices, $lof(\cdot)$. This function maps binary matrices to *left-ordered* binary matrices. $lof(\mathbf{Z})$ is obtained by ordering the columns of the binary matrix $\mathbf{Z}$ from left to right by the magnitude of the binary number expressed by that column, taking the first row as the most significant bit. The left-ordering of a binary matrix is shown in Figure 5. In the first row of the left-ordered matrix, the columns for which $z_{1k} = 1$ are grouped at the left. In the second row, the columns for which $z_{2k} = 1$ are grouped at the left of the sets for which $z_{1k} = 1$. This grouping structure persists throughout the matrix.

Considering the process of placing a binary matrix in left-ordered form motivates the definition of a further technical term. The *history* of feature k at object i is defined to be $(z_{1k}, \ldots, z_{(i-1)k})$. Where no object is specified, we will use *history* to refer to the full history of feature k, $(z_{1k}, \ldots, z_{Nk})$. We will individuate the histories of features using the decimal equivalent of the binary numbers corresponding to the column entries. For example, at object 3, features can have one of four histories: 0, corresponding to a feature with no previous assignments, 1, being a feature for which $z_{2k} = 1$ but $z_{1k} = 0$, 2, being a feature for which $z_{1k} = 1$ but $z_{2k} = 0$, and 3, being a feature possessed by both previous objects were assigned. K_h will denote the number of features possessing the history h, with K_0 being the number of features for which $m_k = 0$ and $K_+ = \sum_{h=1}^{2^N-1} K_h$ being the number of features for which $m_k > 0$, so $K = K_0 + K_+$. The function lof thus places the columns of a matrix in ascending order of their histories.

$lof(\cdot)$ is a many-to-one function: many binary matrices reduce to the same left-ordered form, and there is a unique left-ordered form for every binary matrix. We can thus use $lof(\cdot)$ to define a set of equivalence classes. Any two binary matrices $\mathbf{Y}$ and $\mathbf{Z}$ are lof-equivalent if $lof(\mathbf{Y}) = lof(\mathbf{Z})$, that is, if $\mathbf{Y}$ and $\mathbf{Z}$ map to the same left-ordered form. The lof-equivalence class of a binary matrix $\mathbf{Z}$, denoted $[\mathbf{Z}]$, is the set of binary matrices that are lof-equivalent to $\mathbf{Z}$. lof-equivalence classes

are preserved through permutation of either the rows or the columns of a matrix, provided the same permutations are applied to the other members of the equivalence class. Performing inference at the level of lof-equivalence classes is appropriate in models where feature order is not identifiable, with $p(\mathbf{X}|\mathbf{F})$ being unaffected by the order of the columns of $\mathbf{F}$. Any model in which the probability of $\mathbf{X}$ is specified in terms of a linear function of $\mathbf{F}$, such as PCA or CVQ, has this property.

We need to evaluate the cardinality of $[\mathbf{Z}]$, being the number of matrices that map to the same left-ordered form. The columns of a binary matrix are not guaranteed to be unique: since an object can possess multiple features, it is possible for two features to be possessed by exactly the same set of objects. The number of matrices in $[\mathbf{Z}]$ is reduced if $\mathbf{Z}$ contains identical columns, since some re-orderings of the columns of $\mathbf{Z}$ result in exactly the same matrix. Taking this into account, the cardinality of $[\mathbf{Z}]$ is $\binom{K}{K_0 \ldots K_{2^N-1}} = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!}$, where K_h is the count of the number of columns with full history h.

lof-equivalence classes play the same role for binary matrices as partitions do for assignment vectors: they collapse together all binary matrices (assignment vectors) that differ only in column ordering (class labels). This relationship can be made precise by examining the lof-equivalence classes of binary matrices constructed from assignment vectors. Define the *class matrix* generated by an assignment vector $\mathbf{c}$ to be a binary matrix $\mathbf{Z}$ where $z_{ik} = 1$ if and only if $c_i = k$. It is straightforward to show that the class matrices generated by two assignment vectors that correspond to the same partition belong to the same lof-equivalence class, and vice versa.

4.3 Taking the Infinite Limit

Under the distribution defined by Equation 10, the probability of a particular lof-equivalence class of binary matrices, $[\mathbf{Z}]$, is

$$\begin{aligned} P([\mathbf{Z}]) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} P(\mathbf{Z}) \\ &= \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k+\frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(N+1+\frac{\alpha}{K})}. \end{aligned} \tag{12}$$

In order to take the limit of this expression as $K \to \infty$, we will divide the columns of $\mathbf{Z}$ into two subsets, corresponding to the features for which $m_k = 0$ and the features for which $m_k > 0$. Re-ordering the columns such that $m_k > 0$ if $k \leq K_+$, and $m_k = 0$ otherwise, we can break the product in Equation 12 into two parts, corresponding to these two subsets. The product thus becomes

$$\begin{aligned} &\prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k+\frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(N+1+\frac{\alpha}{K})} \\ &= \left(\frac{\frac{\alpha}{K}\Gamma(\frac{\alpha}{K})\Gamma(N+1)}{\Gamma(N+1+\frac{\alpha}{K})}\right)^{K-K_+} \quad \prod_{k=1}^{K_+} \frac{\frac{\alpha}{K}\Gamma(m_k+\frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(N+1+\frac{\alpha}{K})} \\ &= \left(\frac{\frac{\alpha}{K}\Gamma(\frac{\alpha}{K})\Gamma(N+1)}{\Gamma(N+1+\frac{\alpha}{K})}\right)^{K} \quad \prod_{k=1}^{K_+} \frac{\Gamma(m_k+\frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(\frac{\alpha}{K})\Gamma(N+1)} \\ &= \left(\frac{N!}{\prod_{j=1}^{N}(j+\frac{\alpha}{K})}\right)^{K} \quad \left(\frac{\alpha}{K}\right)^{K_+} \prod_{k=1}^{K_+} \frac{(N-m_k)!\prod_{j=1}^{m_k-1}(j+\frac{\alpha}{K})}{N!}, \end{aligned} \tag{13}$$

where we have used the fact that $\Gamma(x) = (x-1)\Gamma(x-1)$ for $x > 1$. Substituting Equation 13 into Equation 12 and rearranging terms, we can compute our limit

$$\begin{aligned}\lim_{K\to\infty} & \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \cdot \frac{K!}{K_0! K^{K_+}} \cdot \left(\frac{N!}{\prod_{j=1}^{N}(j+\frac{\alpha}{K})}\right)^K \cdot \prod_{k=1}^{K_+} \frac{(N-m_k)!\prod_{j=1}^{m_k-1}(j+\frac{\alpha}{K})}{N!} \\ &= \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \cdot \quad 1 \quad \cdot \quad \exp\{-\alpha H_N\} \quad \cdot \prod_{k=1}^{K_+} \frac{(N-m_k)!(m_k-1)!}{N!}, \end{aligned} \tag{14}$$

where H_N is the Nth harmonic number, $H_N = \sum_{j=1}^{N} \frac{1}{j}$. The details of the steps taken in computing this limit are given in Appendix A. Again, this distribution is exchangeable: neither the number of identical columns nor the column sums are affected by the ordering on objects.

4.4 The Indian Buffet Process

The probability distribution defined in Equation 14 can be derived from a simple stochastic process. As with the CRP, this process assumes an ordering on the objects, generating the matrix sequentially using this ordering. We will also use a culinary metaphor in defining our stochastic process, appropriately adjusted for geography.[4] Many Indian restaurants offer lunchtime buffets with an apparently infinite number of dishes. We can define a distribution over infinite binary matrices by specifying a procedure by which customers (objects) choose dishes (features).

In our Indian buffet process (IBP), N customers enter a restaurant one after another. Each customer encounters a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish, stopping after a Poisson(α) number of dishes as his plate becomes overburdened. The ith customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself with probability $\frac{m_k}{i}$, where m_k is the number of previous customers who have sampled a dish. Having reached the end of all previous sampled dishes, the ith customer then tries a Poisson($\frac{\alpha}{i}$) number of new dishes.

We can indicate which customers chose which dishes using a binary matrix $\mathbf{Z}$ with N rows and infinitely many columns, where $z_{ik} = 1$ if the ith customer sampled the kth dish. Figure 6 shows a matrix generated using the IBP with $\alpha = 10$. The first customer tried 17 dishes. The second customer tried 7 of those dishes, and then tried 3 new dishes. The third customer tried 3 dishes tried by both previous customers, 5 dishes tried by only the first customer, and 2 new dishes. Vertically concatenating the choices of the customers produces the binary matrix shown in the figure.

Using $K_1^{(i)}$ to indicate the number of new dishes sampled by the ith customer, the probability of any particular matrix being produced by this process is

$$P(\mathbf{Z}) = \frac{\alpha^{K_+}}{\prod_{i=1}^{N} K_1^{(i)}!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N-m_k)!(m_k-1)!}{N!}. \tag{15}$$

As can be seen from Figure 6, the matrices produced by this process are generally not in left-ordered form. However, these matrices are also not ordered arbitrarily because the Poisson draws always result in choices of new dishes that are to the right of the previously sampled dishes. Customers are not exchangeable under this distribution, as the number of dishes counted as $K_1^{(i)}$ depends upon

4. This work was started when both authors were at the Gatsby Computational Neuroscience Unit in London, where the Indian buffet is the dominant culinary metaphor.

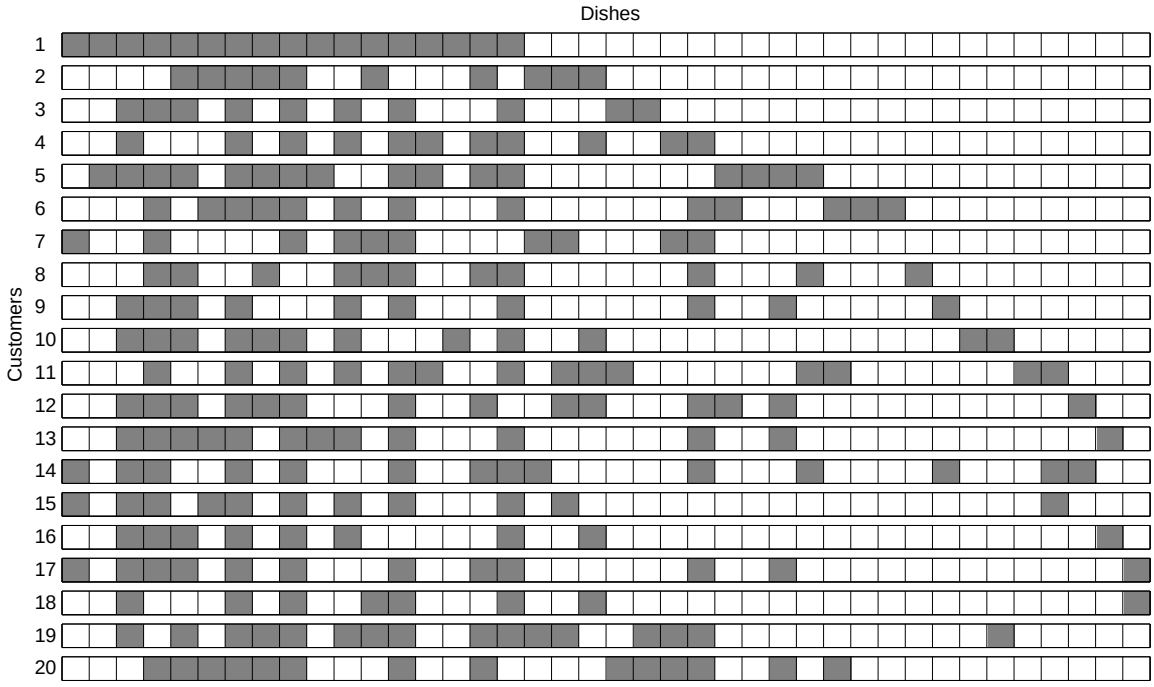

Figure 6: A binary matrix generated by the Indian buffet process with $\alpha = 10$.

the order in which the customers make their choices. However, if we only pay attention to the lof-equivalence classes of the matrices generated by this process, we obtain the exchangeable distribution $P([\mathbf{Z}])$ given by Equation 14: $\frac{\prod_{i=1}^{N} K_1^{(i)}!}{\prod_{h=1}^{2^N-1} K_h!}$ matrices generated via this process map to the same left-ordered form, and $P([\mathbf{Z}])$ is obtained by multiplying $P(\mathbf{Z})$ from Equation 15 by this quantity.

It is possible to define a similar sequential process that directly produces a distribution on lof equivalence classes in which customers are exchangeable, but this requires more effort on the part of the customers. In the *exchangeable* Indian buffet process, the first customer samples a Poisson(α) number of dishes, moving from left to right. The ith customer moves along the buffet, and makes a single decision for each set of dishes with the same history. If there are K_h dishes with history h, under which m_h previous customers have sampled each of those dishes, then the customer samples a Binomial($\frac{m_h}{i}, K_h$) number of those dishes, starting at the left. Having reached the end of all previous sampled dishes, the ith customer then tries a Poisson($\frac{\alpha}{i}$) number of new dishes. Attending to the history of the dishes and always sampling from the left guarantees that the resulting matrix is in left-ordered form, and it is easy to show that the matrices produced by this process have the same probability as the corresponding lof-equivalence classes under Equation 14.

4.5 A Distribution over Collections of Histories

In Section 4.2, we noted that lof-equivalence classes of binary matrices generated from assignment vectors correspond to partitions. Likewise, lof-equivalence classes of general binary matrices correspond to simple combinatorial structures: vectors of non-negative integers. Fixing some ordering of N objects, a collection of feature histories on those objects can be represented by a frequency

vector $\mathbf{K} = (K_1, \ldots, K_{2^N-1})$, indicating the number of times each history appears in the collection. A collection of feature histories can be translated into a left-ordered binary matrix by horizontally concatenating an appropriate number of copies of the binary vector representing each history into a matrix. A left-ordered binary matrix can be translated into a collection of feature histories by counting the number of times each history appears in that matrix. Since partitions are a subset of all collections of histories—namely those collections in which each object appears in only one history—this process is strictly more general than the CRP.

This connection between lof-equivalence classes of feature matrices and collections of feature histories suggests another means of deriving the distribution specified by Equation 14, operating directly on the frequencies of these histories. We can define a distribution on vectors of non-negative integers $\mathbf{K}$ by assuming that each K_h is generated independently from a Poisson distribution with parameter $\alpha B(m_h, N - m_h + 1) = \alpha \frac{(m_h-1)!(N-m_h)!}{N!}$ where m_h is the number of non-zero elements in the history h. This gives

$$\begin{aligned} P(\mathbf{K}) &= \prod_{h=1}^{2^N-1} \frac{\left(\alpha \frac{(m_h-1)!(N-m_h)!}{N!}\right)^{K_h}}{K_h!} \exp\left\{-\alpha \frac{(m_h-1)!(N-m_h)!}{N!}\right\} \\ &= \frac{\alpha^{\sum_{h=1}^{2^N-1} K_h}}{\prod_{h=1}^{2^N-1} K_h!} \exp\{-\alpha H_N\} \prod_{h=1}^{2^N-1} \left(\frac{(m_h-1)!(N-m_h)!}{N!}\right)^{K_h}, \end{aligned}$$

which is easily seen to be the same as $P([\mathbf{Z}])$ in Equation 14. The harmonic number in the exponential term is obtained by summing $\frac{(m_h-1)!(N-m)!}{N!}$ over all histories h. There are $\binom{N}{j}$ histories for which $m_h = j$, so we have

$$\sum_{h=1}^{2^N-1} \frac{(m_h-1)!(N-m_h)!}{N!} = \sum_{j=1}^{N} \binom{N}{j} \frac{(j-1)!(N-j)!}{N!} = \sum_{j=1}^{N} \frac{1}{j} = H_N. \tag{16}$$

4.6 Properties of this Distribution

These different views of the distribution specified by Equation 14 make it straightforward to derive some of its properties. First, the effective dimension of the model, K_+, follows a Poisson(αH_N) distribution. This is easily shown using the generative process described in Section 4.5: $K_+ = \sum_{h=1}^{2^N-1} K_h$, and under this process is thus the sum of a set of Poisson distributions. The sum of a set of Poisson distributions is a Poisson distribution with parameter equal to the sum of the parameters of its components. Using Equation 16, this is αH_N. Alternatively, we can use the fact that the number of new columns generated at the ith row is Poisson($\frac{\alpha}{i}$), with the total number of columns being the sum of these quantities.

A second property of this distribution is that the number of features possessed by each object follows a Poisson(α) distribution. This follows from the definition of the exchangeable IBP. The first customer chooses a Poisson(α) number of dishes. By exchangeability, all other customers must also choose a Poisson(α) number of dishes, since we can always specify an ordering on customers which begins with a particular customer.

Finally, it is possible to show that $\mathbf{Z}$ remains sparse as $K \to \infty$. The simplest way to do this is to exploit the previous result: if the number of features possessed by each object follows a Poisson(α) distribution, then the expected number of entries in $\mathbf{Z}$ is $N\alpha$. This is consistent with the quantity

obtained by taking the limit of this expectation in the finite model, which is given in Equation 11: $\lim_{K\to\infty} E\left[\mathbf{1}^T\mathbf{Z}\mathbf{1}\right] = \lim_{K\to\infty} \frac{N\alpha}{1+\frac{\alpha}{K}} = N\alpha$.

4.7 Inference by Gibbs Sampling

We have defined a distribution over infinite binary matrices that satisfies one of our desiderata—objects (the rows of the matrix) are exchangeable under this distribution. It remains to be shown that inference in infinite latent feature models is tractable, as was the case for infinite mixture models. We will derive a Gibbs sampler for sampling from the distribution defined by the IBP, which suggests a strategy for inference in latent feature models in which the exchangeable IBP is used as a prior. We will consider alternative inference algorithms later in the paper.

To sample from the distribution defined by the IBP, we need to compute the conditional distribution $P(z_{ik} = 1|\mathbf{Z}_{-(ik)})$, where $\mathbf{Z}_{-(ik)}$ denotes the entries of $\mathbf{Z}$ other than z_{ik}. In the finite model, where $P(\mathbf{Z})$ is given by Equation 10, it is straightforward to compute the conditional distribution for any z_{ik}. Integrating over π_k gives

$$\begin{aligned} P(z_{ik} = 1|\mathbf{z}_{-i,k}) &= \int_0^1 P(z_{ik}|\pi_k)\, p(\pi_k|\mathbf{z}_{-i,k})\, d\pi_k \\ &= \frac{m_{-i,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}, \end{aligned} \tag{17}$$

where $\mathbf{z}_{-i,k}$ is the set of assignments of other objects, not including i, for feature k, and $m_{-i,k}$ is the number of objects possessing feature k, not including i. We need only condition on $\mathbf{z}_{-i,k}$ rather than $\mathbf{Z}_{-(ik)}$ because the columns of the matrix are generated independently under this prior.

In the infinite case, we can derive the conditional distribution from the exchangeable IBP. Choosing an ordering on objects such that the ith object corresponds to the last customer to visit the buffet, we obtain

$$P(z_{ik} = 1|\mathbf{z}_{-i,k}) = \frac{m_{-i,k}}{N}, \tag{18}$$

for any k such that $m_{-i,k} > 0$. The same result can be obtained by taking the limit of Equation 17 as $K \to \infty$. Similarly the number of new features associated with object i should be drawn from a Poisson($\frac{\alpha}{N}$) distribution. This can also be derived from Equation 17, using the same kind of limiting argument as that presented above to obtain the terms of the Poisson.

This analysis results in a simple Gibbs sampling algorithm for generating samples from the distribution defined by the IBP. We start with an arbitrary binary matrix. We then iterate through the rows of the matrix, i. For each column k, if $m_{-i,k}$ is greater than 0 we set $z_{ik} = 1$ with probability given by Equation 18. Otherwise, we delete that column. At the end of the row, we add Poisson($\frac{\alpha}{N}$) new columns that have ones in that row. After sufficiently many passes through the rows, the resulting matrix will be a draw from the distribution $P(\mathbf{Z})$ given by Equation 15.

This algorithm suggests a heuristic strategy for sampling from the posterior distribution $P(\mathbf{Z}|\mathbf{X})$ in a model that uses the IBP to define a prior on $\mathbf{Z}$. In this case, we need to sample from the full conditional distribution

$$P(z_{ik} = 1|\mathbf{Z}_{-(ik)}, \mathbf{X}) \propto p(\mathbf{X}|\mathbf{Z})P(z_{ik} = 1|\mathbf{Z}_{-(ik)})$$

where $p(\mathbf{X}|\mathbf{Z})$ is the likelihood function for the model, and we assume that parameters of the likelihood have been integrated out. We can proceed as in the Gibbs sampler given above, simply

incorporating the likelihood term when sampling z_{ik} for columns for which $m_{-i,k}$ is greater than 0 and drawing the new columns from a distribution where the prior is Poisson($\frac{\alpha}{N}$) and the likelihood is given by $P(\mathbf{X}|\mathbf{Z})$.[5]

5. An Example: A Linear-Gaussian Latent Feature Model with Binary Features

We have derived a prior for infinite sparse binary matrices, and indicated how statistical inference can be done in models defined using this prior. In this section, we will show how this prior can be put to use in models for unsupervised learning, illustrating some of the issues that can arise in this process. We will describe a simple linear-Gaussian latent feature model, in which the features are binary. As above, we will start with a finite model and then consider the infinite limit.

5.1 A Finite Linear-Gaussian Model

In our finite model, the D-dimensional vector of properties of an object i, $\mathbf{x}_i$ is generated from a Gaussian distribution with mean $\mathbf{z}_i\mathbf{A}$ and covariance matrix $\Sigma_X = \sigma_X^2\mathbf{I}$, where $\mathbf{z}_i$ is a K-dimensional binary vector, and $\mathbf{A}$ is a $K \times D$ matrix of weights. In matrix notation, $E[\mathbf{X}] = \mathbf{ZA}$. If $\mathbf{Z}$ is a feature matrix, this is a form of binary factor analysis. The distribution of $\mathbf{X}$ given $\mathbf{Z}$, $\mathbf{A}$, and σ_X is matrix Gaussian:

$$p(\mathbf{X}|\mathbf{Z},\mathbf{A},\sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{ND/2}} \exp\{-\frac{1}{2\sigma_X^2}\mathrm{tr}((\mathbf{X}-\mathbf{ZA})^T(\mathbf{X}-\mathbf{ZA}))\} \tag{19}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix. This makes it easy to integrate out the model parameters $\mathbf{A}$. To do so, we need to define a prior on $\mathbf{A}$, which we also take to be matrix Gaussian:

$$p(\mathbf{A}|\sigma_A) = \frac{1}{(2\pi\sigma_A^2)^{KD/2}} \exp\{-\frac{1}{2\sigma_A^2}\mathrm{tr}(\mathbf{A}^T\mathbf{A})\}, \tag{20}$$

where σ_A is a parameter setting the diffuseness of the prior. The dependencies among the variables in this model are shown in Figure 7.

Combining Equations 19 and 20 results in an exponentiated expression involving the trace of

$$\begin{aligned}
&\frac{1}{\sigma_X^2}(\mathbf{X}-\mathbf{ZA})^T(\mathbf{X}-\mathbf{ZA}) + \frac{1}{\sigma_A^2}\mathbf{A}^T\mathbf{A} \\
&\qquad = \frac{1}{\sigma_X^2}\mathbf{X}^T\mathbf{X} - \frac{1}{\sigma_X^2}\mathbf{X}^T\mathbf{ZA} - \frac{1}{\sigma_X^2}\mathbf{A}^T\mathbf{Z}^T\mathbf{X} + \mathbf{A}^T(\frac{1}{\sigma_X^2}\mathbf{Z}^T\mathbf{Z} + \frac{1}{\sigma_A^2}\mathbf{I})\mathbf{A} \\
&\qquad = \frac{1}{\sigma_X^2}(\mathbf{X}^T(\mathbf{I}-\mathbf{ZMZ}^T)\mathbf{X}) + (\mathbf{MZ}^T\mathbf{X}-\mathbf{A})^T(\sigma_X^2\mathbf{M})^{-1}(\mathbf{MZ}^T\mathbf{X}-\mathbf{A}),
\end{aligned}$$

5. As was pointed out by an anonymous reviewer, this is a heuristic strategy rather than a valid algorithm for sampling from the posterior because it violates one of the assumptions of Markov chain Monte Carlo algorithms, with the order in which variables are sampled being dependent on the state of the Markov chain. This is not an issue in the algorithm for sampling from $P(\mathbf{Z})$, since the columns of $\mathbf{Z}$ are independent, and the kernels corresponding to sampling from each of the conditional distributions thus act independently of one another.

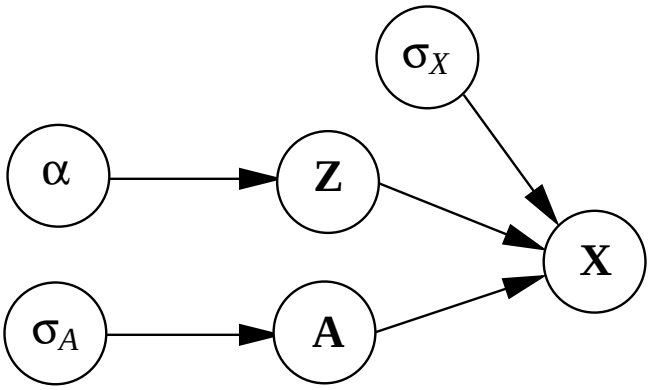

Figure 7: Graphical model for the linear-Gaussian model with binary features.

where $\mathbf{I}$ is the identity matrix, $\mathbf{M} = (\mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}$, and the last line is obtained by completing the square for the quadratic term in $\mathbf{A}$ in the second line. We can then integrate out $\mathbf{A}$ to obtain

$$\begin{aligned}
&p(\mathbf{X}|\mathbf{Z},\sigma_X,\sigma_A) \\
&\quad= \int p(\mathbf{X}|\mathbf{Z},\mathbf{A},\sigma_X)p(\mathbf{A}|\sigma_A)\,d\mathbf{A} \\
&\quad= \frac{1}{(2\pi)^{(N+K)D/2}\sigma_X^{ND}\sigma_A^{KD}}\exp\{-\frac{1}{2\sigma_X^2}\mathrm{tr}(\mathbf{X}^T(\mathbf{I}-\mathbf{Z}\mathbf{M}\mathbf{Z}^T)\mathbf{X})\} \\
&\qquad\qquad \int \exp\{-\frac{1}{2}\mathrm{tr}((\mathbf{M}\mathbf{Z}^T\mathbf{X}-\mathbf{A})^T(\sigma_X^2\mathbf{M})^{-1}(\mathbf{M}\mathbf{Z}^T\mathbf{X}-\mathbf{A}))\}\,d\mathbf{A} \\
&\quad= \frac{|\sigma_X^2\mathbf{M}|^{D/2}}{(2\pi)^{ND/2}\sigma_X^{ND}\sigma_A^{KD}}\exp\{-\frac{1}{2\sigma_X^2}\mathrm{tr}(\mathbf{X}^T(\mathbf{I}-\mathbf{Z}\mathbf{M}\mathbf{Z}^T)\mathbf{X})\} \\
&\quad= \frac{1}{(2\pi)^{ND/2}\sigma_X^{(N-K)D}\sigma_A^{KD}|\mathbf{Z}^T\mathbf{Z}+\frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}|^{D/2}} \\
&\qquad\qquad \exp\{-\frac{1}{2\sigma_X^2}\mathrm{tr}(\mathbf{X}^T(\mathbf{I}-\mathbf{Z}(\mathbf{Z}^T\mathbf{Z}+\frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\mathbf{Z}^T)\mathbf{X})\}. \qquad (21)
\end{aligned}$$

This result is intuitive: the exponentiated term is the difference between the inner product matrix of the raw values of $\mathbf{X}$ and their projections onto the space spanned by $\mathbf{Z}$, regularized to an extent determined by the ratio of the variance of the noise in $\mathbf{X}$ to the variance of the prior on $\mathbf{A}$. This is simply the marginal likelihood for a Bayesian linear regression model (Minka, 2000).

We can use this derivation of $p(\mathbf{X}|\mathbf{Z},\sigma_X,\sigma_A)$ to infer $\mathbf{Z}$ from a set of observations $\mathbf{X}$, provided we have a prior on $\mathbf{Z}$. The finite feature model discussed as a prelude to the IBP is such a prior. The full conditional distribution for z_{ik} is given by:

$$P(z_{ik}|\mathbf{X},\mathbf{Z}_{-(i,k)},\sigma_X,\sigma_A) \propto p(\mathbf{X}|\mathbf{Z},\sigma_X,\sigma_A)P(z_{ik}|\mathbf{z}_{-i,k}). \qquad (22)$$

While evaluating $p(\mathbf{X}|\mathbf{Z},\sigma_X,\sigma_A)$ always involves matrix multiplication, it need not always involve a matrix inverse. $\mathbf{Z}^T\mathbf{Z}$ can be rewritten as $\sum_i \mathbf{z}_i^T\mathbf{z}_i$, allowing us to use rank one updates to efficiently

compute the inverse when only one $\mathbf{z}_i$ is modified. Defining $\mathbf{M}_{-i} = (\sum_{j \neq i} \mathbf{z}_j^T \mathbf{z}_j + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}$, we have

$$\begin{aligned} \mathbf{M}_{-i} &= (\mathbf{M}^{-1} - \mathbf{z}_i^T \mathbf{z}_i)^{-1} \\ &= \mathbf{M} - \frac{\mathbf{M}\mathbf{z}_i^T \mathbf{z}_i \mathbf{M}}{\mathbf{z}_i \mathbf{M} \mathbf{z}_i^T - 1}, \qquad (23) \\ \mathbf{M} &= (\mathbf{M}_{-i}^{-1} + \mathbf{z}_i^T \mathbf{z}_i)^{-1} \\ &= \mathbf{M}_{-i} - \frac{\mathbf{M}_{-i}\mathbf{z}_i^T \mathbf{z}_i \mathbf{M}_{-i}}{\mathbf{z}_i \mathbf{M}_{-i} \mathbf{z}_i^T + 1}. \qquad (24) \end{aligned}$$

Iteratively applying these updates allows $p(\mathbf{X}|\mathbf{Z}, \sigma_X, \sigma_A)$, to be computed via Equation 21 for different values of z_{ik} without requiring an excessive number of inverses, although a full rank update should be made occasionally to avoid accumulating numerical errors. The second part of Equation 22, $P(z_{ik}|\mathbf{z}_{-i,k})$, can be evaluated using Equation 17.

5.2 Taking the Infinite Limit

To make sure that we can define an infinite version of this model, we need to check that $p(\mathbf{X}|\mathbf{Z}, \sigma_X, \sigma_A)$ remains well-defined if $\mathbf{Z}$ has an unbounded number of columns. $\mathbf{Z}$ appears in two places in Equation 21: in $|\mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}|$ and in $\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\mathbf{Z}^T$. We will examine how these behave as $K \to \infty$.

If $\mathbf{Z}$ is in left-ordered form, we can write it as $[\mathbf{Z}_+ \ \mathbf{Z}_0]$, where $\mathbf{Z}_+$ consists of K_+ columns with sums $m_k > 0$, and $\mathbf{Z}_0$ consists of K_0 columns with sums $m_k = 0$. It follows that the first of the two expressions we are concerned with reduces to

$$\begin{aligned} \left| \mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I} \right| &= \left| \begin{bmatrix} \mathbf{Z}_+^T\mathbf{Z}_+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}_K \right| \\ &= \left(\frac{\sigma_X^2}{\sigma_A^2}\right)^{K_0} \left| \mathbf{Z}_+^T\mathbf{Z}_+ + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}_{K_+} \right|. \qquad (25) \end{aligned}$$

The appearance of K_0 in this expression is not a problem, as we will see shortly. The abundance of zeros in $\mathbf{Z}$ leads to a direct reduction of the second expression to

$$\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\mathbf{Z}^T = \mathbf{Z}_+(\mathbf{Z}_+^T\mathbf{Z}_+ + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}_{K_+})^{-1}\mathbf{Z}_+^T,$$

which only uses the finite portion of $\mathbf{Z}$. Combining these results yields the likelihood for the infinite model

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}, \sigma_X, \sigma_A) &= \frac{1}{(2\pi)^{ND/2}\sigma_X^{(N-K_+)D}\sigma_A^{K_+D}|\mathbf{Z}_+^T\mathbf{Z}_+ + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}_{K_+}|^{D/2}} \\ &\quad \exp\{-\frac{1}{2\sigma_X^2}\mathrm{tr}(\mathbf{X}^T(\mathbf{I} - \mathbf{Z}_+(\mathbf{Z}_+^T\mathbf{Z}_+ + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}_{K_+})^{-1}\mathbf{Z}_+^T)\mathbf{X})\}. \qquad (26) \end{aligned}$$

The K_+ in the exponents of σ_A and σ_X appears as a result of introducing $D/2$ multiples of the factor of $\left(\frac{\sigma_X^2}{\sigma_A^2}\right)^{K_0}$ from Equation 25. The likelihood for the infinite model is thus just the likelihood for the finite model defined on the first K_+ columns of $\mathbf{Z}$.

The heuristic Gibbs sampling algorithm defined in Section 4.7 can now be used in this model. Assignments to classes for which $m_{-i,k} > 0$ are drawn in the same way as for the finite model, via Equation 22, using Equation 26 to obtain $p(\mathbf{X}|\mathbf{Z}, \sigma_X, \sigma_A)$ and Equation 18 for $P(z_{ik}|\mathbf{z}_{-i,k})$. As in the finite case, Equations 23 and 24 can be used to compute inverses efficiently. The distribution over the number of new features can be approximated by truncation, computing probabilities for a range of values of $K_1^{(i)}$ up to some reasonable upper bound. For each value, $p(\mathbf{X}|\mathbf{Z}, \sigma_X, \sigma_A)$ can be computed from Equation 26, and the prior on the number of new classes is Poisson($\frac{\alpha}{N}$). More elaborate samplers which do not require truncation are presented in Meeds et al. (2007) and in Teh et al. (2007).

5.3 Demonstrations

As a first demonstration of the ability of this algorithm to recover the latent structure responsible for having generated observed data, we applied the Gibbs sampler for the infinite linear-Gaussian model to a simulated data set consisting of 100 6×6 images, each generated by randomly assigning a feature to each image to a class with probability 0.5, and taking a linear combination of the weights associated with features to which the images were assigned (a similar data set was used by Ghahramani, 1995). Some of these images are shown in Figure 8, together with the weights $\mathbf{A}$ that were used to generate them. The non-zero elements of $\mathbf{A}$ were all equal to 1.0, and σ_X was set to 0.5, introducing a large amount of noise.

The algorithm was initialized with $K_+ = 1$, choosing the feature assignments for the first column by setting $z_{i1} = 1$ with probability 0.5. σ_A was set to 1.0. The Gibbs sampler rapidly discovered that four classes were sufficient to account for the data, and converged to a distribution focused on matrices $\mathbf{Z}$ that closely matched the true class assignments. The results are shown in Figure 8. Each of the features is represented by the posterior mean of the feature weights, $\mathbf{A}$, given $\mathbf{X}$ and $\mathbf{Z}$, which is

$$E[\mathbf{A}|\mathbf{X}, \mathbf{Z}] = (\mathbf{Z}^T\mathbf{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\mathbf{Z}^T\mathbf{X}.$$

for a single sample $\mathbf{Z}$. The results shown in the figure are from the 200th sample produced by the algorithm.

These results indicate that the algorithm can recover the features used to generate simulated data. In a further test of the algorithm with more realistic data, we applied it to a data set consisting of 100 240×320 pixel images. We represented each image, $\mathbf{x}_i$, using a 100-dimensional vector corresponding to the weights of the mean image and the first 99 principal components. Each image contained up to four everyday objects—a $20 bill, a Klein bottle, a prehistoric handaxe, and a cellular phone. The objects were placed in fixed locations, but were put into the scenes by hand, producing some small variation in location. The images were then taken with a low resolution webcam. Each object constituted a single latent feature responsible for the observed pixel values. The images were generated by sampling a feature vector, $\mathbf{z}_i$, from a distribution under which each feature was present with probability 0.5, and then taking a photograph containing the appropriate objects using a LogiTech digital webcam. Sample images are shown in Figure 9 (a). The only noise in the images was the noise from the camera.

The Gibbs sampler was initialized with $K_+ = 1$, choosing the feature assignments for the first column by setting $z_{i1} = 1$ with probability 0.5. σ_A, σ_X, and α were initially set to 0.5, 1.7, and 1 respectively, and then sampled by adding Metropolis steps to the MCMC algorithm. Figure 9

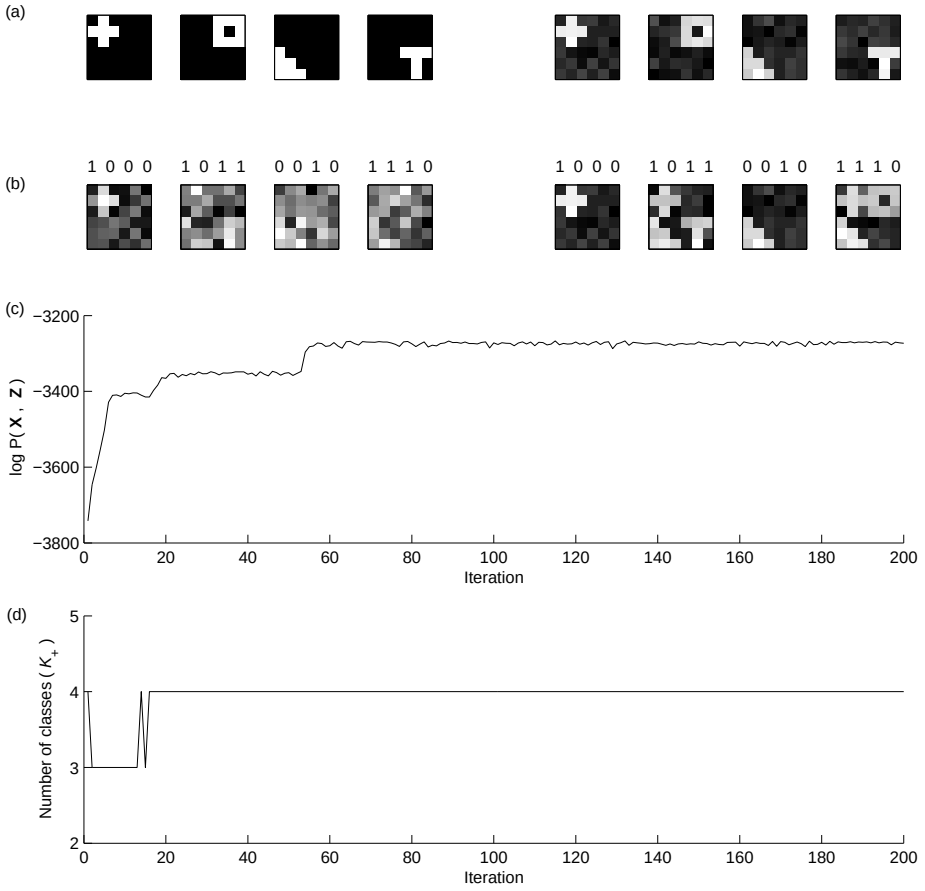

Figure 8: Demonstration of the linear-Gaussian model described in the text, using a binary representation. (a) 100 images were generated as binary linear combinations of four sets of class weights, shown in the images on the left. The images on the right are the posterior mean weights **A** for a single sample of **Z** after 200 iterations, ordered to match the true classes. (b) The images on the left show four of the datapoints to which the model was applied. The numbers above each image indicate the classes responsible for generating that image, matching the order above. The images on the right show the predictions of the model for these images, based on the posterior mean weights, together with the class assignments from the sampled **Z**. (c) Trace plot of $\log P(\mathbf{X}, \mathbf{Z})$ over 200 iterations. (d) Trace plot of K_+, the number of classes, over 200 iterations. The data were generated from a model with $K_+ = 4$.

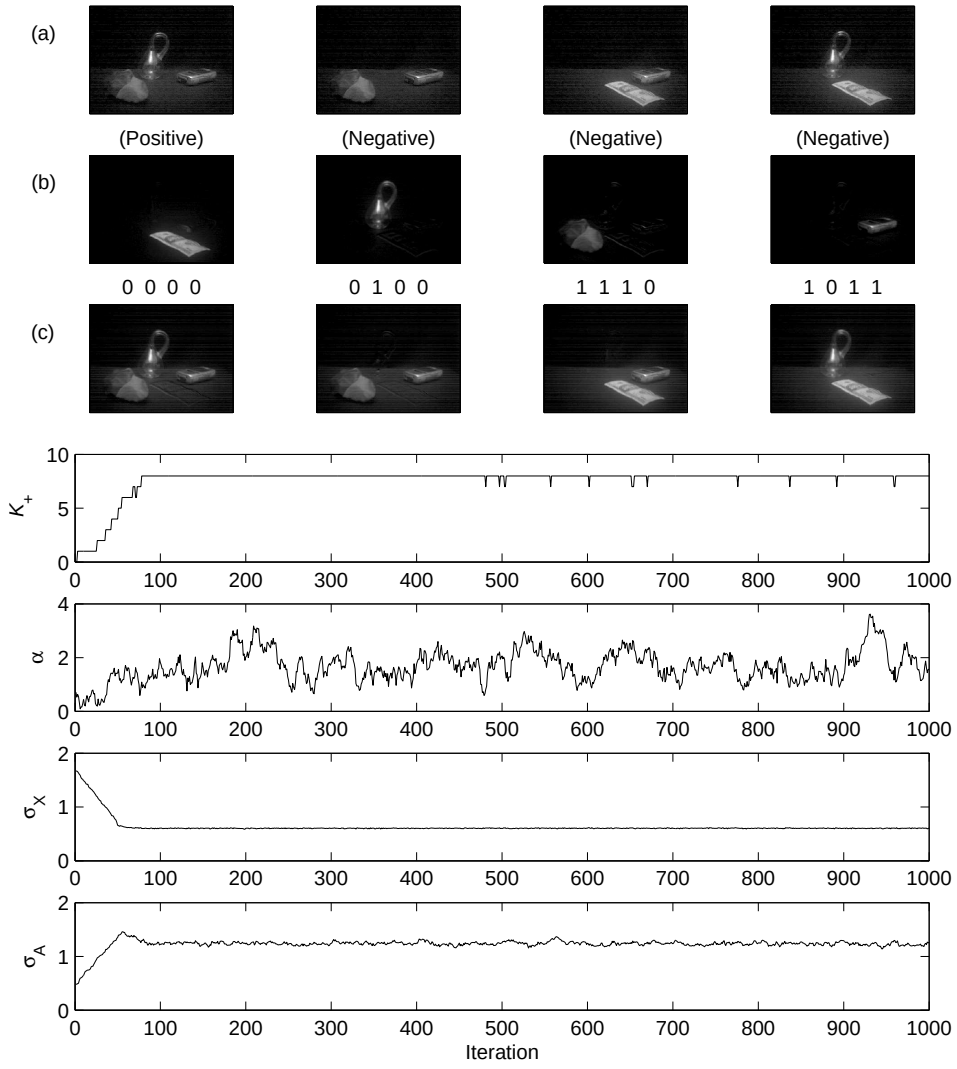

Figure 9: Data and results for the application of the infinite linear-Gaussian model to photographic images. (a) Four sample images from the 100 in the data set. Each image had 320×240 pixels, and contained from zero to four everyday objects. (b) The posterior mean of the weights (**A**) for the four most frequent binary features from the 1000th sample. Each image corresponds to a single feature. These features perfectly indicate the presence or absence of the four objects. The first feature indicates the presence of the \$20 bill, the other three indicate the absence of the Klein bottle, the handaxe, and the cellphone. (c) Reconstructions of the images in (a) using the binary codes inferred for those images. These reconstructions are based upon the posterior mean of **A** for the 1000th sample. For example, the code for the first image indicates that the \$20 bill is absent, while the other three objects are not. The lower panels show trace plots for the dimensionality of the representation (K_+) and the parameters α, σ_X, and σ_A over 1000 iterations of sampling. The values of all parameters stabilize after approximately 100 iterations.

shows trace plots for the first 1000 iterations of MCMC for the number of features used by at least one object, K_+, and the model parameters σ_A, σ_X, and α. All of these quantities stabilized after approximately 100 iterations, with the algorithm finding solutions with approximately seven latent features.

Figure 9 (b) shows the posterior mean of $\mathbf{a}_k$ for the four most frequent features in the 1000th sample produced by the algorithm. These features perfectly indicated presence and absence of the four objects. Three less common features coded for slight differences in the locations of those objects. Figure 9 (c) shows the feature vectors $\mathbf{z}_i$ from this sample for the four images in Figure 9(b), together with the posterior means of the reconstructions of these images for this sample, $\mathbf{z}_i E[\mathbf{A}|\mathbf{X},\mathbf{Z}]$. Similar reconstructions are obtained by averaging over all values of $\mathbf{Z}$ produced by the Markov chain. The reconstructions provided by the model clearly pick out the relevant content of the images, removing the camera noise in the original images.

These applications of the linear-Gaussian latent feature model are intended primarily to demonstrate that this nonparametric Bayesian approach can efficiently learn satisfying representations without requiring the dimensionality of those representations to be fixed a priori. The data set consisting of images of objects was constructed in a way that removes many of the basic challenges of computer vision, with objects appearing in fixed orientations and locations. Dealing with these issues requires using a more sophisticated image representation or a more complex likelihood function than the linear-Gaussian model. Despite its simplicity, the example of identifying the objects in images illustrates the kind of problems for which the IBP provides an appropriate prior. We describe a range of other applications of the Indian buffet process in detail in the next section.

6. Further Applications and Alternative Inference Algorithms

We now outline six applications of the Indian buffet process, each of which uses the same prior over infinite binary matrices, $P(\mathbf{Z})$, but different choices for the likelihood relating such matrices to observed data. These applications provide an indication of the potential uses of the IBP in machine learning, and have also led to a number of alternative inference algorithms, which we will describe briefly.

6.1 Choice Behavior

Choice behavior refers to our ability to decide between several options. Models of choice behavior are of interest to psychology, marketing, decision theory, and computer science. Our choices are often governed by features of the different options. For example, when choosing which car to buy, one may be influenced by fuel efficiency, cost, size, make, etc. Görür et al. (2006) present a nonparametric Bayesian model based on the IBP which, given the choice data, infers latent features of the options and the corresponding weights of these features. The likelihood function is taken from Tversky's (1972) classic "elimination by aspects" model of choice, with the probability of choosing option A over option B being proportional to the sum of the weights of the distinctive features of A. The IBP is the prior over these latent features, which are assumed to be either present or absent.

The likelihood function used in this model does not have a natural conjugate prior, meaning that the approach taken in our Gibbs sampling algorithm—integrating out the parameters associated with the features—cannot be used. This led Görür et al. to develop a similar Markov chain Monte Carlo algorithm for use with a non-conjugate prior. The basic idea behind the algorithm is analogous to Algorithm 8 of Neal (2000) for Dirichlet process mixture models, using a set of auxiliary variables

to represent the weights associated with features that are currently not possessed by any of the available options. These auxiliary variables effectively provide a Monte Carlo approximation to the sum over parameters used in our Gibbs sampler (although there is no approximation error introduced through this step).

6.2 Modeling Protein Interactions

Proteomics aims to understand the functional interactions of proteins, and is a field of growing importance to modern biology and medicine. One of the key concepts in proteomics is a *protein complex*, a group of several interacting proteins. Protein complexes can be experimentally determined by doing high-throughput protein-protein interaction screens. Typically the results of such experiments are subjected to mixture-model based clustering methods. However, a protein can belong to multiple complexes at the same time, making the mixture model assumption invalid. Chu et al. (2006) proposed a nonparametric Bayesian approach based on the IBP for identifying protein complexes and their constituents from interaction screens. The latent binary feature z_{ik} indicates whether protein i belongs to complex k. The likelihood function captures the probability that two proteins will be observed to bind in the interaction screen as a function of how many complexes they both belong to, $\sum_{k=1}^{\infty} z_{ik} z_{jk}$. The approach automatically infers the number of significant complexes from the data and the results are validated using affinity purification/mass spectrometry experimental data from yeast RNA-processing complexes.

6.3 Binary Matrix Factorization for Modeling Dyadic Data

Many interesting data sets are *dyadic*: there are two sets of objects or entities and observations are made on pairs with one element from each set. For example, the two sets might consist of movies and viewers, and the observations are ratings given by viewers to movies. Alternatively, the two sets might be genes and biological tissues and the observations may be expression levels for particular genes in different tissues. Dyadic data can often be represented as matrices and many models of dyadic data can be expressed in terms of matrix factorization. Models of dyadic data make it possible to predict, for example, the ratings a viewer might give to a movie based on ratings from other viewers, a task known as *collaborative filtering*. A traditional approach to modeling dyadic data is *bi-clustering*: simultaneously clustering both the rows (e.g., viewers) and the columns (e.g., movies) of the observation matrix using coupled mixture models. However, as we have discussed, mixture models provide a very limited latent variable representation of data. Meeds et al. (2007) presented a more expressive model of dyadic data based on the two-parameter version of the Indian buffet process. In this model, both movies and viewers are represented by binary latent vectors with an unbounded number of elements, corresponding to the features they might possess (e.g., "likes horror movies"). The two corresponding infinite binary matrices interact via a real-valued weight matrix which links features of movies to features of viewers, resulting in a binary matrix factorization of the observed ratings.

The basic inference algorithm used in this model was similar to the non-conjugate version of the Gibbs sampler outlined above, but the authors also developed a number of novel Metropolis-Hastings proposals that are mixed with the steps of the Gibbs sampler. One proposal directly handles the number of new features associated with each object, facilitating one of the more difficult aspects of non-conjugate inference. Another proposal is a "split-merge" move, analogous to similar proposals used in models based on the CRP (Jain and Neal, 2004; Dahl, 2003). In contrast to the

Gibbs sampler, which slowly affects the number of features used in the model by changing a single feature allocation for a single object at a time, the split-merge proposal explores large-scale moves such as dividing a single feature into two, or collapsing two features together. Combining these large-scale moves with the Gibbs sampler can result in a Markov chain Monte Carlo algorithm that explores the space of latent matrices faster.

6.4 Extracting Features from Similarity Judgments

One of the goals of cognitive psychology is to determine the kinds of representations that underlie people's judgments. In particular, the *additive clustering* method has been used to infer people's beliefs about the features of objects from their judgments of the similarity between them (Shepard and Arabie, 1979). Given a square matrix of judgments of the similarity between N objects, where s_{ij} is the similarity between objects i and j, the additive clustering model seeks to recover a $N \times K$ binary feature matrix $\mathbf{F}$ and a vector of K weights associated with those features such that $s_{ij} \approx \sum_{k=1}^{K} w_k f_{ik} f_{jk}$. A standard problem for this approach is determining the value of K, for which a variety of heuristic methods have been used. Navarro and Griffiths (2007) presented a nonparametric Bayesian solution to this problem, using the IBP to define a prior on $\mathbf{F}$ and assuming that s_{ij} has a Gaussian distribution with mean $\sum_{k=1}^{K_+} w_k f_{ik} f_{jk}$ (following Tenenbaum, 1996). Using this method provides a posterior distribution over the effective dimension of $\mathbf{F}$, K_+, and gives both a weight and a posterior probability for the presence of each feature.

Samples from the posterior distribution over feature matrices reveal some surprisingly rich representations expressed in classic similarity data sets. Performing posterior inference makes it possible to discover that there are multiple sensible sets of features that could account for human similarity judgments, while previous approaches that had focused on finding the single best set of features might only find one such set. For example, the nonparametric Bayesian model reveals that people's similarity judgments for numbers from 0-9 can be accounted for by a set of features that includes both the odd and the even numbers, while previous additive clustering analyses (e.g., Tenenbaum, 1996) had only produced the odd numbers.

The additive clustering model, like the choice model discussed above, is another case in which non-conjugate inference is necessary. In this case, the inference algorithm is rendered simpler by the fact that no attempt is made to model the similarity of an object to itself, s_{ii}. As a consequence, a feature possessed by a single object has no effect on the likelihood, and the number of such features and their associated weights can be drawn directly from the prior. Inference thus proceeds using an algorithm similar to the Gibbs sampler derived above, with the addition of a Metropolis-Hastings step to update the weights associated with each feature.

6.5 Latent Features in Link Prediction

Network data, indicating the relationships among a group of people or objects, have been analyzed by both statisticians and sociologists. A basic goal of these analyses is predicting which unobserved relationships might exist. For example, having observed friendly interactions among several pairs of people, a sociologist might seek to predict which other people are likely to be friends with one another. This problem of link prediction can be solved using a probabilistic model for the structure of graphs. One popular class of models, known as stochastic blockmodels, assume that each entity belongs to a single latent class, and that the probability of a relationship existing between two entities depends only on the classes of those entities (Nowicki and Snijders, 2001; Wang and Wong,

1987). This is analogous to a mixture model, in which the probability that an object has certain observed properties depends only on its latent class. Nonparametric versions of stochastic blockmodels can be defined using the Chinese restaurant process (Kemp et al., 2006), corresponding to an underlying stochastic process that generalizes the Dirichlet process (Roy and Teh, 2009).

Just as allowing objects to have latent features rather than a single latent class makes it possible to go beyond mixture models, this approach allows us to define models for link prediction that are richer than stochastic blockmodels. Miller et al. (2010) defined a class of nonparametric latent feature models that can be used for link prediction. The key idea is to define the probability of the existence of a link between two entities in terms of a "squashing function" (such as the logistic or probit) applied to a real-valued score for that link. The scores then depend on the features of the two entities. For a set of N entities, the pairwise scores are given by the $N \times N$ matrix $\mathbf{ZWZ}^T$, where $\mathbf{Z}$ is a binary feature matrix, as used throughout this paper, and $\mathbf{W}$ is a matrix of real-valued feature weights. Since the feature weights can be positive or negative, features can interact to either increase or decrease the probability of a link. The resulting model is strictly more expressive than a stochastic blockmodel and produces more accurate predictions, particularly in cases where multiple factors interact to influence the existence of a relationship (such as in the decision to co-author a paper, for example).

6.6 Independent Components Analysis and Sparse Factor Analysis

Independent Components Analysis (ICA) is a model which explains observed signals in terms of a linear superposition, or mixing, of independent hidden sources (Comon, 1994; Bell and Sejnowski, 1995; MacKay, 1996; Cardoso, 1998). ICA has been used to solve the problem of "blind source separation" in which the goal is to unmix the hidden sources from the observed mixed signals without assuming much knowledge of the hidden source distribution. This models, for example, a listener in a cocktail party who may want to unmix the signals received on his two ears into the many independent sound sources that produced them. ICA is closely related to factor analysis, except that while in factor analysis the sources are assumed to be Gaussian distributed, in ICA the sources are assumed to have any distribution other than the Gaussian.

One of the key practical problems in ICA is determining the number of hidden sources. Knowles and Ghahramani (2007) provided a solution to this problem by devising a non-parametric Bayesian model for ICA based on the IBP. The basic assumption of this ICA model is that the number of potential sources is unbounded, but that any particular source is typically not present in a given signal. The IBP provides a natural model for determining which sources are present in each signal. In the notation of Section 3, the observed signals are represented by a matrix $\mathbf{X}$, the presence or absence of the hidden sources by the IBP distributed matrix $\mathbf{Z}$, and the value taken by the sources by the matrix $\mathbf{V}$. Knowles and Ghahramani (2007) considered several variants of the model, including ICA models where the elements of $\mathbf{V}$ have Laplacian distributions, sparse FA models where the elements of $\mathbf{V}$ have Gaussian distributions, and one and two parameter versions of the IBP in both cases. The model was applied to discovering gene signatures from gene expression microarray data from an ovarian cancer study.

Rai and Daumé (2009) developed two interesting extensions of this model also motivated by applications to gene expression data. First they considered both factor analysis and factor regression models, where the latter refers to solving a regression problem with a typically large number of input features by making predictions based solely on the factor representation. Second, they used

an IBP to model the sparsity in the factor loading matrix (rather than the factor or source matrix in nonparametric ICA) and they moreover assume that the factors are related to each other through a hierarchy. They used Kingman's coalescent as a nonparametric Bayesian model for this hierarchy, following the inference algorithms developed in Teh et al. (2008). This paper shows a nice example of how the IBP can be integrated with other nonparametric Bayesian distributions in a fairly modular manner to solve useful inference problems.

6.7 Bipartite Graphs and Learning Hidden Causes

Wood et al. (2006) used the IBP as part of an algorithm for learning the structure of graphical models. Specifically, they focused on the case where an unknown number of hidden variables (e.g., diseases) are causes for some set of observed variables (e.g., symptoms). Rather than defining a prior over the number of hidden causes, Wood et al. used a non-parametric Bayesian approach based on the IBP to model the structure of graphs with countably infinitely many hidden causes. The binary variable z_{ik} indicates whether hidden variable k has a direct causal influence on observed variable i; in other words whether k is a parent of i in the graph. The data being modeled were the values of the set of observed variables over a number of trials, where each variable was either present or absent on each trial. Each hidden variable could be either present or absent on a particular trial, with the probabilities of these states being determined by a parameter of the model, and hidden variables were assumed to combine via a noisy-OR (Pearl, 1988) to influence the observed variables.

Wood et al. (2006) described an MCMC algorithm for inference in this model. Like many of the cases discussed in this section, this model lacked natural conjugate priors. Inference was done using a variant on the Gibbs sampler introduced above, with additional steps to modify the values of the hidden variables. The sampling step for the introduction of new hidden causes into the graph was facilitated by an analytic result making it possible to sum out the values of the variables associated with those causes in a way that is analogous to summing out the parameters in a conjugate model. However, Wood and Griffiths (2007) developed a sequential Monte Carlo algorithm for use in this model, similar to algorithms that have been developed for use with the CRP (such as Fearnhead, 2004). This algorithm is a form of particle filter, updating the posterior distribution on $\mathbf{Z}$ one row at a time (in this case, as new observed variables are added to the data). The particle filter provides an efficient and straightforward alternative for inference in models that lack conjugate priors, and generalizes naturally to other models using the IBP.

6.8 Structuring Markov Transition Matrices

Discrete Markov processes are widely used in machine learning, as part of hidden Markov models and state-space models. Nonparametric Bayesian methods have been used to define "infinite" versions of these models, allowing the number of states in a hidden Markov model to be unbounded (Beal et al., 2002). An infinite discrete Markov process can be defined by assuming that transitions from each state follow a Chinese restaurant process, with transitions that have been made frequently in the past being more likely in the future. When a new transition is generated, the next state is drawn from a higher-level Chinese restaurant process that is shared across all states. The resulting distribution can also be obtained from a hierarchical Dirichlet process (Teh et al., 2004).

Fox et al. (2010) recently explored another way of defining an infinite discrete Markov process, which allows for more structure in the transition matrix. In this model, it is assumed that each state can only make transitions to a subset of other states. Thus, each state is associated with a binary

vector indicating whether or not it makes transitions to other states. With an infinite set of states, a distribution over these vectors can be defined using the IBP. This approach was used to define a nonparametric autoregressive hidden Markov model, in which a sequence of continuous variables were predicted as a linear function of the variables at the previous timestep, but the parameters of the function were determined by a latent Markov process. The resulting model was able to identify meaningful action components in motion capture data. In addition to introducing a novel model, this paper explored the use of "birth and death" moves in the Markov chain Monte Carlo algorithm used for inference, in which entire columns of the matrix produced by the IBP were created or destroyed.

6.9 Other Inference Algorithms

The broad range of settings in which the IBP has been applied have encouraged the development of more efficient methods for probabilistic inference in the resulting nonparametric Bayesian models. As discussed above, several innovations have been used to speed mixing in the Markov chain Monte Carlo algorithms used with specific models. Other work has explored schemes for making inference in the linear-Gaussian model discussed in Section 5 more efficient and scalable to larger data sets. For example, if instead of integrating out the weight matrix $\mathbf{A}$, the posterior distribution over $\mathbf{A}$ is maintained, it is possible to use an alternative sampling scheme that still mixes quickly where the time for each iteration scales linearly in N (Doshi-Velez and Ghahramani, 2009a). This observation also provides the basis for a parallelization scheme in which the features of different objects are computed on different machines, with the potential to make large-scale applications of this linear-Gaussian model possible (Doshi-Velez et al., 2010). Similar principles may apply in the other models using the IBP discussed in this section.

An alternative approach to probabilistic inference is to reject the stochastic approximations provided by MCMC algorithms in favor of deterministic approximations, using variational inference to approximate the posterior. A mean field approximation to the IBP was developed by Doshi-Velez et al. (2009), building on similar approximations for Dirichlet process mixture models (Blei and Jordan, 2006). This variational inference method was applied to the infinite ICA model discussed in Section 6.6, and compared against sampling schemes on both synthetic and real data. The results of these comparisons suggested that the variational approach provides a more efficient strategy for inference in this model when the dimensionality of the observed data is high. Variational inference may thus be useful in working with some of the other models discussed in this section, at least in specific regimes.

7. Extensions and Connections to Other Processes

The Indian buffet process gives a way to characterize our distribution on infinite binary matrices in terms of a simple stochastic process. In this section we review how the IBP can be extended to yield more general classes of distributions, and summarize some of the connections between the IBP and other stochastic processes. Our derivation of the IBP was based on considering the infinite limit of a distribution on finite binary matrices. As with the CRP, this distribution can also be derived via a stick-breaking construction, or by marginalizing out an underlying measure. These different views of the IBP yield different generalizations of the distribution, and different opportunities for developing inference algorithms.

7.1 A Two-Parameter Generalization

As was discussed in Section 4.6, the distribution on the number of features per object and on the total number of features produced by the IBP are directly coupled, through α. This is an undesirable constraint, as the sparsity of a matrix and its dimensionality should be able to vary independently. Ghahramani et al. (2007) introduced a two-parameter generalization of the IBP that separates these two aspects of the distribution.[6] This generalization keeps the average number of features per object at α as before, but allows the overall number of represented features to range from α, an extreme where all features are shared between all objects, to $N\alpha$, an extreme where no features are shared at all. Between these extremes lie many distributions that capture the amount of sharing appropriate for different domains.

As the one-parameter model, this two-parameter model can be derived by taking the limit of a finite model, but using $\pi_k|\alpha,\beta \sim \text{Beta}(\frac{\alpha\beta}{K},\beta)$ instead of Equation 9. Here we will focus on the equivalent sequential generative process. To return to the language of the Indian buffet, the first customer starts at the left of the buffet and samples Poisson(α) dishes. The ith customer serves himself from any dish previously sampled by $m_k > 0$ customers with probability $m_k/(\beta+i-1)$, and in addition from Poisson$(\alpha\beta/(\beta+i-1))$ new dishes. The parameter β is introduced in such a way as to preserve the expected number of features per object, α, but the expected overall number of features is $\alpha\sum_{i=1}^{N}\frac{\beta}{\beta+i-1}$, and the distribution of K_+ is Poisson with this mean. The total number of features used thus increases as β increases. For finite β, the expected number of features increases as $\alpha\beta\ln N$, but if $\beta \gg 1$ the logarithmic regime is preceded by linear growth at small $N < \beta$.

Figure 10 shows three matrices drawn from the two-parameter IBP, all with $\alpha = 10$ but with $\beta = 0.2$, $\beta = 1$, and $\beta = 5$ respectively. Although all three matrices have roughly the same number of non-zero entries, the number of features used varies considerably. At small values of β features become likely to be shared by all objects. At high values of β features are more likely to be specific to particular objects. Further details about the properties of this distribution are provided in Ghahramani et al. (2007).

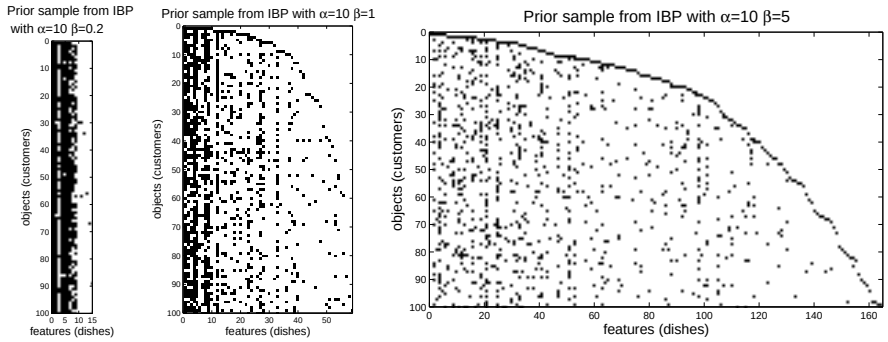

Figure 10: Three samples from the two-parameter Indian buffet process with $\alpha = 10$ and $\beta = 0.2$ (left), $\beta = 1$ (middle), and $\beta = 5$ (right).

7.2 A Stick-Breaking Construction

Our strategy of taking the limit of a finite exchangeable distribution in deriving the IBP was inspired by the derivation of the CRP as the limit of a Dirichlet-multinomial model. However, there are many

6. The original idea and analysis was described in an unpublished note by Sollich (2005).

other routes by which the CRP can be derived. One of these is via the Dirichlet process (Ferguson, 1973). A simple way to think about the Dirichlet process is in terms of a probability measure over probability measures. The parameters of the process are its concentration α and a base measure G_0. In a typical use, we would draw a measure G from the Dirichlet process, and then generate parameters for a model ϕ_i by sampling them independently from G. Since the Dirichlet process generates discrete measures with probability 1, it is possible for multiple parameters ϕ_i and ϕ_j drawn from G to take the same value. We can thus imagine indexing the values taken by the ϕ_i with discrete variables z_i, such that $z_i = z_j$ if and only if $\phi_i = \phi_j$. The z_i thus index unique values of ϕ_i, and correspond to a partition of the indices of the ϕ_i. The distribution over partitions $\mathbf{z}$ produced by the Dirichlet process, integrating over G, is the CRP (Blackwell and MacQueen, 1973).

A straightforward way to understand how the Dirichlet process allocates probabilities to a discrete set of atoms is to think about assigning probabilities in terms of breaking off pieces of a stick. The stick is one unit in length, corresponding to the fact that our probabilities must sum to one. Each piece of stick we break off represents the probability assigned to another discrete atom. After breaking off each piece, we then consider how much of the remainder to break off as the next piece. Sethuraman (1994) showed that if this process is repeated infinitely often, with a proportion of the stick drawn from a Beta$(\alpha, 1)$ distribution being broken off at each step, the lengths of the pieces of broken stick are equivalent to the probabilities assigned to a discrete set of atoms by the Dirichlet process with parameter α. This stick-breaking representation of the Dirichlet process is useful in deriving its properties, and in developing inference algorithms such as the variational inference algorithm proposed by Blei and Jordan (2006).

Teh et al. (2007) showed that a similar stick-breaking construction can be defined for the IBP. First, we imagine sorting the π_k representing the probability of each feature being possessed by an object from largest to smallest. Then, if we consider the proportion of the stick that is broken off and discarded at each break in the stick-breaking construction for the Dirichlet process, the distribution of the sequence of stick lengths corresponds exactly to the distribution of these ordered probabilities. This stick-breaking construction identifies an interesting relationship between the IBP and the Dirichlet process, and is useful for exactly the same reasons. In particular, the stick-breaking construction was used in defining the variational inference algorithm summarized in Section 6.9, and can also be used to derive other inference algorithms for the IBP, such as slice sampling (Teh et al., 2007).

7.3 Connections to the Beta Process

The relationship between the CRP and the Dirichlet process is an instance of a more general relationship between exchangeable distributions and underlying probability measures. The results summarized in the previous paragraph indicate that we can write

$$P(\mathbf{z}) = \int \prod_{i=1}^{N} P(z_i|G)\, p(G)\, dG,$$

where the z_i are drawn independently from the measure G, which is generated from the Dirichlet process. The fact that we can represent the exchangeable distribution $P(\mathbf{z})$ as the result of generating the z_i independently from a latent measure is a specific instance of the more general principle stated in de Finetti's exchangeability theorem, which indicates that *any* exchangeable distribution can be represented in this way (see Bernardo and Smith, 1994, for details). This raises a natural question: is there a similar measure underlying the exchangeable distribution produced by the IBP?

Thibaux and Jordan (2007) provided an answer to this question, showing that the exchangeable distribution produced by the IBP corresponds to the use of a latent measure based on the beta process (Hjort, 1990). The beta process provides a source of Bernoulli parameters π_k associated with the elements of a (possibly continuous) index set. Sampling each of the z_{ik} independently according to the distribution defined by the appropriate parameter results in the same distribution on $\mathbf{Z}$ as the IBP. This perspective also makes it straightforward to define analogues of the two-parameter process described in Section 7.1, and to extend the IBP to a hierarchical model that can capture correlations in the features exhibited in multiple data sets. Teh and Görür (2010) also recently used the relationship to the beta process to define a variant of the IBP that produces a power-law distribution in feature frequencies, exploiting a connection to stable processes. Variants of this kind may be useful in settings where power-law distributions are common, such as natural language processing.

7.4 Relaxing the Assumption of Exchangeability

The IBP assumes independence between the columns of $\mathbf{Z}$, and only the kind of weak dependency implied by exchangeability for the rows of $\mathbf{Z}$. Both of these assumptions have been relaxed in subsequent work. Producing correlations between the columns of $\mathbf{Z}$ can be done by supplementing the IBP with a secondary process capturing patterns in the latent features (Doshi-Velez and Ghahramani, 2009b). Modifying the assumption of exchangeability is potentially more problematic. Exchangeability was one of our original desiderata, since it is a reasonable assumption in many settings and simplifies probabilistic inference. However, this assumption is not warranted in cases where we have additional information about the properties of our observations, such as the fact that they were produced in a particular temporal sequence, or reflect a known pattern of correlation. The challenge is thus to identify how the assumption of exchangeability can be relaxed while maintaining the tractability of probabilistic inference. Two recent papers have presented strategies for modifying the IBP to capture different forms of dependency between the rows of $\mathbf{Z}$.

The first kind of dependency can arise as the consequence of observations being generated in a specific sequence. In such a case, it might be appropriate to assume that the latent features associated with observations made closer in time should be more correlated. A strategy for modifying the IBP to capture this kind of dependency was introduced by Van Gael et al. (2009). In this model—the Markov Indian buffet process—it is assumed that the rows of $\mathbf{Z}$ are generated via a Markov process, where the values in each column are generated based on the corresponding values in the previous row. This Markov process has two parameters, giving the probability of a 0 in the previous row changing to a 1, and the probability of a 1 in the previous row remaining unchanged. By assuming that these parameters are generated from a Beta distribution and taking a limit analogous to that used in the derivation of the IBP, it is possible to define a distribution over equivalence classes of binary matrices in which the rows of the matrix reflect a Markov dependency structure. This model can be used to define richer nonparametric models for temporal data, such as an infinite factorial hidden Markov model, and probabilistic inference can be carried out using a slice sampler (see Van Gael et al., 2009, for details).

A second kind of dependency can be the result of known degrees of relatedness among observations. For example, one might seek to draw inferences about a group of people with known genetic relationships, or about a set of organisms or languages with a known evolutionary history. In cases where the degrees of relatedness can be expressed in a tree, the phylogenetic Indian buffet process

(Miller et al., 2008) can be used. In this model, the tree expresses the dependency structure that governs the rows of $\mathbf{Z}$, and each column is generated independently by sampling from a stochastic process defined on the tree. The parameters of the stochastic process are specified in a way that guarantees the total number of columns follows a Poisson distribution, and the original IBP is recovered as the special case where the tree is degenerate, with all branches meeting at the root. Trees can be used to capture a wide range of dependency structures, including partial exchangeability, and probabilistic inference by MCMC remains tractable because belief propagation on the tree can be used to efficiently compute the relevant conditional probabilities.

8. Conclusions and Future Work

The methods that have been used to define infinite latent class models can be extended to models in which objects are represented in terms of a set of latent features, and used to derive distributions on infinite binary matrices that can be used as priors for such models. We used this method to derive a prior that is the infinite limit of a simple distribution on finite binary matrices, and showed that the same distribution can be specified in terms of a simple stochastic process—the Indian buffet process. This distribution satisfies our two desiderata for a prior for infinite latent feature models: objects are exchangeable, and inference remains tractable. When used as a prior in models that represent objects using latent features, this distribution can be used to automatically infer the number of features required to account for observed data. More generally, it can be used as a prior in any setting where a sparse binary matrix with a finite number of rows and infinite number of columns is appropriate, such as estimating the adjacency matrix of a bipartite graph where the size of one class of nodes is unknown.

Recent work has made significant progress on turning this nonparametric approach to inferring latent features into a tool that can be used to solve a wide range of machine learning problems. These advances include more sophisticated MCMC algorithms, schemes for parallelizing probabilistic inference, and deterministic methods for approximating posterior distributions over latent feature matrices. The connections between the IBP and other stochastic processes provide the groundwork for further understanding and extending this class of probabilistic models, making it possible to modify the distribution over feature assignments and to capture different patterns of dependency that might exist among the latent features of objects. As with the CRP, the different views of the IBP that result from considering the stick-breaking construction or the underlying measure that is marginalized out to obtain the combinatorial stochastic process each support different extensions, generalizations, and inference algorithms.

Despite the wide array of successful applications of the IBP and related distributions, we view one of the primary contributions of this work to be the idea that we can define richer nonparametric Bayesian models to suit the unique challenges of machine learning. Our success in transferring the strategy of taking the limit of a finite model from latent classes to latent features suggests that the same strategy might be applied with other representations, broadening the kinds of latent structure that can be recovered through unsupervised learning. This idea receives support both from other examples of new nonparametric models defined via a similar strategy (e.g., Titsias, 2008), and from theoretical analyses of the conditions under which infinite models remain well defined when obtained as limits of finite models (Orbanz, 2010). We anticipate that there will be other combinatorial structures for which this strategy will result in new and useful distributions.

Acknowledgments

This work was presented at the Neural Information Processing Systems conference, and draws on the conference paper (Griffiths and Ghahramani, 2006) and associated technical report (Griffiths and Ghahramani, 2005). The preparation of this article was supported by grants BCS-0631518 and IIS-0845410 from the National Science Foundation, and grant FA-9550-10-1-0232 from the Air Force Office of Scientific Research. We thank three anonymous reviewers for their comments on the manuscript.

Appendix A. Details of Limits

This appendix contains the details of the limits of three expressions that appear in Equations 5 and 14.

The first expression is

$$\begin{aligned}\frac{K!}{K_0!K^{K_+}} &= \frac{\prod_{k=1}^{K_+}(K-k+1)}{K^{K_+}} \\ &= \frac{K^{K_+} - \frac{(K_+-1)K_+}{2}K^{K_+-1} + \cdots + (-1)^{K_+-1}(K_+-1)!K}{K^{K_+}} \\ &= 1 - \frac{(K_+-1)K_+}{2K} + \cdots + \frac{(-1)^{K_+-1}(K_+-1)!}{K^{K_+-1}}.\end{aligned}$$

For finite K_+, all terms except the first go to zero as $K \to \infty$.

The second expression is

$$\prod_{j=1}^{m_k-1}(j+\tfrac{\alpha}{K}) = (m_k-1)! + \tfrac{\alpha}{K}\sum_{j=1}^{m_k-1}\frac{(m_k-1)!}{j} + \cdots + \left(\tfrac{\alpha}{K}\right)^{m_k-1}.$$

For finite m_k and α, all terms except the first go to zero as $K \to \infty$.

The third expression is

$$\begin{aligned}\left(\frac{N!}{\prod_{j=1}^{N}(j+\frac{\alpha}{K})}\right)^K &= \left(\frac{\prod_{j=1}^{N} j}{\prod_{j=1}^{N}(j+\frac{\alpha}{K})}\right)^K \\ &= \left(\prod_{j=1}^{N}\frac{j}{(j+\frac{\alpha}{K})}\right)^K \\ &= \prod_{j=1}^{N}\left(\frac{1}{1+\frac{\alpha\frac{1}{j}}{K}}\right)^K. \qquad (27)\end{aligned}$$

We can now use the fact that

$$\lim_{K\to\infty}\left(\frac{1}{1+\frac{x}{K}}\right)^K = \exp\{-x\}$$

to compute the limit of Equation 27 as $K \to \infty$, obtaining

$$\begin{aligned}\lim_{K\to\infty} \prod_{j=1}^{N} \left(\frac{1}{1+\frac{\alpha\frac{1}{j}}{K}} \right)^{K} &= \prod_{j=1}^{N} \exp\{-\alpha\tfrac{1}{j}\} \\ &= \exp\{-\alpha \sum_{j=1}^{N} \tfrac{1}{j}\} \\ &= \exp\{-\alpha H_N\},\end{aligned}$$

as desired.

References

D. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, pages 1–198. Springer, Berlin, 1985.

C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174, 1974.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. *Machine Learning*, pages 29–245, 2002.

A. J. Bell and T. J. Sejnowski. An information maximisation approach to blind separation and blind deconv olution. *Neural Computation*, 7(6):1129–1159, 1995.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, New York, 1994.

D. Blackwell and J. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

D. Blei and M. Jordan. Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, 1:121–144, 2006.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

C. A. Bush and S. N. MacEachern. A semi-parametric Bayesian model for randomized block designs. *Biometrika*, 83:275–286, 1996.

J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10): 2009–2025, Oct 1998.

W. Chu, Z. Ghahramani, R. Krause, and D. L. Wild. Identifying protein complexes in high-throughput protein interaction screens using an infinite latent feature model. In *BIOCOMPUTING 2006: Proceedings of the Pacific Symposium*, volume 11, pages 231–242, 2006.

P. Comon. Independent component analysis: A new concept. *Signal Processing*, 36:287–314, 1994.

D. B. Dahl. An improved merge-split sampler for conjugate Dirichlet process mixture models. Technical Report 1086, Department of Statistics, University of Wisconsin, 2003.

A. d'Aspremont, L. El Ghaoui, I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. Technical Report UCB/CSD-04-1330, Computer Science Division, University of California, Berkeley, 2004.

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

F. Doshi-Velez and Z. Ghahramani. Correlated non-parametric latent feature models. In *Proceedings of the Proceedings of the Twenty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 143–150, 2009b.

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

F. Doshi-Velez, D. Knowles, S. Mohamed, and Z. Ghahramani. Large scale nonparametric Bayesian inference: Data parallelisation in the Indian buffet process. In *Advances in Neural Information Processing Systems 22*, 2010.

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

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

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

T. S. Ferguson. Bayesian density estimation by mixtures of normal distributions. In M. Rizvi, J. Rustagi, and D. Siegmund, editors, *Recent Advances in Statistics*, pages 287–302. Academic Press, New York, 1983.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems 22*, 2010.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Z. Ghahramani. Factorial learning and the EM algorithm. In *Advances in Neural Information Processing Systems 7*. Morgan Kaufmann, San Francisco, CA, 1995.

Z. Ghahramani, T. L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. In *Bayesian Statistics 8*. Oxford University Press, Oxford, 2007.

W.R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk, UK, 1996.

D. Görür, F. Jäkel, and C. E. Rasmussen. A choice model with infinitely many latent features. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pages 361–368, New York, 2006. ACM Press.

P. Green and S. Richardson. Modelling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics*, 28:355–377, 2001.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. Technical Report 2005-001, Gatsby Computational Neuroscience Unit, 2005.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2006. MIT Press.

K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *International Conference on Machine Learning (ICML 2005)*, 2005.

N. L. Hjort. Nonparametric Bayes estimators based on Beta processes in models for life history data. *Annals of Statistics*, 18:1259–1294, 1990.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96:1316–1332, 2001.

S. Jain and R. M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet Process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004.

I. T. Jolliffe. *Principal component analysis*. Springer, New York, 1986.

I. T. Jolliffe and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *7th International Conference on Independent Component Analysis and Signal Separation (ICA 2007)*, Lecture Notes in Computer Science Series (LNCS). Springer, 2007.

D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Technical Report Draft 3.7, Cavendish Laboratory, University of Cambridge, Madingley Road, Cambridge CB3 0HE, December 1996.

E. Meeds, Z. Ghahramani, R. Neal, and S. T. Roweis. Modeling dyadic data with binary latent factors. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA, 2007. MIT Press.

K. T. Miller, T. L. Griffiths, and M. I. Jordan. The phylogenetic Indian buffet process: A non-exchangeable nonparametric prior for latent features. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI 2008)*, 2008.

K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link predictions. In *Advances in Neural Information Processing Systems 22*, 2010.

T. Minka. Bayesian linear regression. Technical report, MIT Media Lab, 2000. http://research.microsoft.com/en-us/um/people/minka/papers/linear.html.

D. J. Navarro and T. L. Griffiths. A nonparametric Bayesian model for inferring features from similarity judgments. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, Cambridge, MA, 2007. MIT Press.

R. M. Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, pages 197–211. Kluwer, Dordrecht, 1992.

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

R. M. Neal. Density modeling and clustering using dirichlet diffusion trees. In J. M. Bernardo et al., editor, *Bayesian Statistics 7*, pages 619–629, 2003.

K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

P. Orbanz. Construction of nonparametric Bayesian models from parametric Bayes equations. In *Advances in Neural Information Processing Systems 22*, 2010.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, CA, 1988.

J. Pitman. Combinatorial stochastic processes, 2002. Notes for Saint Flour Summer School.

P. Rai and H. Daumé. The infinite hierarchical factor regression model. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

C. Rasmussen. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge, MA, 2000.

C. E. Rasmussen and Z. Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge, MA, 2001.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

D. M. Roy and Y. W. Teh. The mondrian process. In *Advances in Neural Information Processing Systems 21*, 2009.

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

R. Shepard and P. Arabie. Additive clutering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86:87–123, 1979.

P. Sollich. Indian buffet process with tunable feature repulsion, 2005.

E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed Dirichlet processes. In *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2006. MIT Press.

Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2004.

Y. W. Teh and D. Görür. Indian buffet processes with power-law behavior. In *Advances in Neural Information Processing Systems 22*, 2010.

Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, San Juan, Puerto Rico, 2007.

Y. W. Teh, H. Daumé, and D. M. Roy. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

J. B. Tenenbaum. Learning the structure of similarity. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in neural information processing systems 8*, pages 3–9. MIT Press, Cambridge, MA, 1996.

R. Thibaux and M. I. Jordan. Hierarchical Beta processes and the Indian buffet process. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, 2007.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

M. Titsias. The infinite gamma-poisson feature model. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

A. Tversky. Elimination by aspects: A theory of choice. *Psychological Review*, 79:281–299, 1972.

N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, Cambridge, 2003. MIT Press.

J. Van Gael, Y.W. Teh, and Z. Ghahramani. The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

M. West, P. Muller, and M. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In P. Freeman and A. Smith, editors, *Aspects of Uncertainty*, pages 363–386. Wiley, New York, 1994.

F. Wood and T. L. Griffiths. Particle filtering for nonparametric Bayesian matrix factorization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1513–1520. MIT Press, Cambridge, MA, 2007.

F. Wood, T. L. Griffiths, and Z. Ghahramani. A non-parametric Bayesian method for inferring hidden causes. In *Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence (UAI '06)*, 2006.

R. S. Zemel and G. E. Hinton. Developing population codes by minimizing description length. In *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann, San Francisco, CA, 1994.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:262–286, 2006.