10-708: Probabilistic Graphical Models, Spring 2015

19 : Bayesian Nonparametrics: The Indian Buffet Process

Lecturer: Avinava Dubey *Scribes: Rishav Das, Adam Brodie, and Hemank Lamba*

1 Latent Variable Models and the Indian Buffet Process

1.1 Latent Variable Models

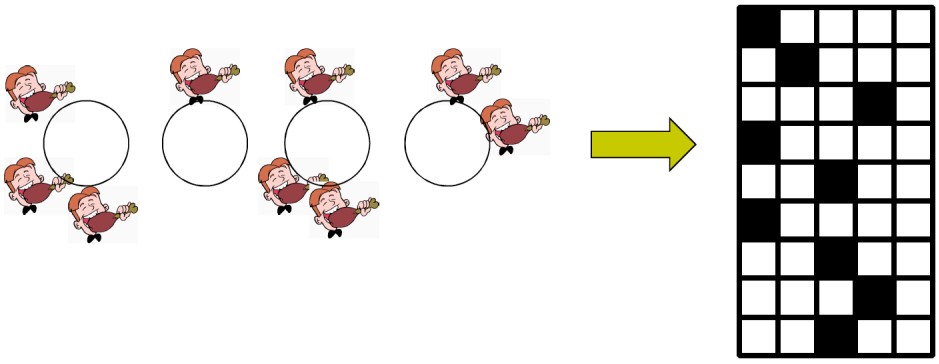
In the previous lecture we discussed Dirichlet processes and the Chinese Restaurant metaphor. A Dirichlet process is a generative model which describes a discrete mixture distribution with infinitely many mixture components, the name referring to the Dirichlet distribution over component weights. Dirichlet processes are useful for unsupervised clustering applications but do not afford the representation of compositional latent structure; each cluster is atomic and has no internal structure as would be ideal for representing objects with a multiplicity of latent features.

A model class which is better suited to such problems are latent feature models; examples include factor analysis (lecture 11) and latent Dirichlet allocation or other topics models (lecture 15). A latent feature model takes the general form,

$$X = WA^{\intercal} + \epsilon$$

where X denotes the observable random vector, W denotes a matrix of weights specific to each datum, A denotes a matrix of latent features, and ϵ denotes independent noise.

A problem in latent feature modeling is selecting the number of latent features to use. This is analogous to the problem that arises in clustering of selecting the number of clusters to include in a model. Dirichlet processes provided a solution in the clustering domain by allowing for infinitely many clusters and letting the data decide how many clusters are actually used to describe the data generating process. A similar approach can be applied to latent feature models by allowing for infinitely many latent features. The consequence is that require W to have infinitely many columns (A to have infinitely many rows). This may be made tractable by requiring that W be sparse so that for any particular model, only finitely many features are used in the description of any data set. This already occurs naturally in Dirichlet processes; each datum is assigned to a single cluster, representing cluster membership as latent features the number of features required to describe N data is bounded by N.



1.2 Sparse Latent Feature Models

We can utilize a strategy parallel to that used in the construction of the Dirichlet process to define a latent feature model which incorporates infinitely many latent features, enforces sparsity sufficient for proper definition, and relaxes the constraint of the Dirichlet process that each item have a non-zero weight for only a single feature. We first examine the finite case in order to develop the intuitions for how this occurs.

Just as we used a Dirichlet prior over cluster parameters in the Dirichlet process we may use a beta prior over Bernoulli parameters for latent feature values in a sparse latent feature model. Suppose we have K many features. For each k from 1 to K and for each n from 1 to N,

$$\begin{aligned} \pi_k &\sim \text{Beta}\left(\frac{\alpha}{K}, 1\right) \\ w_{n,k} &\sim \text{Bernoulli}(\pi_k) \end{aligned}$$

where here $w_{n,k}$ denotes a dummy variable indicating whether the nth item satisfies the kth feature. The probability of some configuration of W and π, the vector of all π_k, can then be expressed,

$$p(W, \pi) = \prod_{k=1}^{K} \prod_{n=1}^{N} \text{Beta}\left(\pi_k; \frac{\alpha}{K}, 1\right) \pi_k^{w_{n,k}}$$

Marginalizing out π,

$$\begin{aligned} p(W) &= \prod_{k=1}^{K} \int \prod_{n=1}^{N} \text{Beta}\left(\pi_k; \frac{\alpha}{K}, 1\right) \pi_k^{w_{n,k}} d_\pi \\ &= \prod_{k=1}^{K} \frac{B(m_k + \alpha K^{-1}, N - m_k + 1)}{B(\alpha K^{-1}, 1)} \\ &= \prod_{k=1}^{K} \frac{(\alpha K^{-1})\Gamma(m_k + \alpha K^{-1})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \alpha K^{-1})} \end{aligned}$$

where m_k denotes the number of items satisfying feature k, i.e., $m_k = \sum_{n+1}^{N} w_{n,k}$. We may observe that the

expected density of W is bounded by the choice of α.

$$\begin{aligned} E(1^{\intercal}W1) &= N^{-1}\sum_{k=1}^{K}\sum_{n=1}^{N} w_{n,k} \\ &= NK\frac{\alpha K^{-1}}{1+\alpha K^{-1}} \\ &= \frac{N\alpha}{1+\alpha K^{-1}} \end{aligned}$$

Since features are exchangeable, each matrix W is an instance of an equivalence class of identical latent feature models which we denote $[W]$. We need to compute the size of each equivalence class in order to compute $p([W])$, the probability of the latent feature model. We compute the size of $[W]$ as follows. Let us call $h = (w_{1,k}, \ldots, w_{n-1,k})$ the *history* of feature k at item n (note that h is an integer expressed in binary notation). Let K_h denote the number of features k with history h and let K_+ denote the total number of features with non-zero histories. The size of $[K]$ may then be expressed as,

$$\binom{K}{\prod_{i=0}^{2^N-1} K_i} = \frac{K!}{\prod_{i=0}^{2^N-1} K_i!}$$

Using this expression we can then express the probability of $[W]$.

$$\begin{aligned} p([W]) &= \sum_{W\in[W]} p(W) \\ &= \frac{K!}{\prod_{i=0}^{2^N-1} K_i!}\prod_{k=1}^{K}\frac{(\alpha K^{-1})\Gamma(m_k+\alpha K^{-1})\Gamma(N-m_k+1)}{\Gamma(N+1+\alpha K^{-1})} \\ &= \frac{K!\alpha^{K_+}}{\left(\prod_{i=0}^{2^N-1} K_i!\right) K_n!(K_0!)K^{K_+}}\left(\frac{N!}{\prod_{j=1}^{N} j+\alpha K^{-1}}\right)^{K}\prod_{k=1}^{K_+}\frac{(N-m_k)!\prod_{l=1}^{m_k-1}(l+\alpha K^{-1})}{N!} \end{aligned}$$

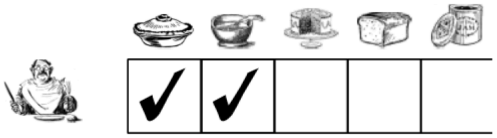
1.3 The Indian Buffet Process

We now consider our Beta-Bernoulli model as K, the number of latent features, goes to the limit.

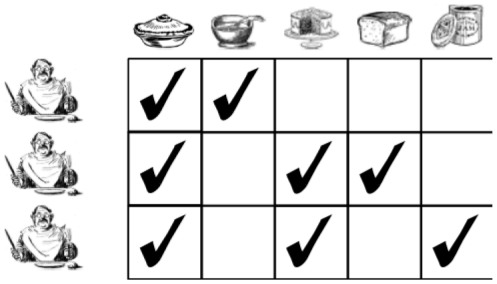
$$\lim_{K\to\infty} p([W]) = \frac{\alpha^{K_+}}{\left(\prod_{i=0}^{2^N-1} K_i!\right) K_n!}\exp\left(-\alpha\sum_{j=1}^{N} j^{-1}\right)\prod_{k=1}^{K_+}\frac{(N-m_k)!(m_k-1)!}{N!}$$

This describes the *Indian Buffet Process*. Sampling from such a process can be described by metaphor in a fashion analogous to the Chinese Restaurant metaphor used to describe sampling from a Dirichlet process. The metaphor goes as follows.

There's an Indian buffet restaurant with infinitely many dishes along the buffet aligned in a sequence. One customer enters the restaurant to begin the process. She helps herself to some of each of the first a many dishes along the buffet with $a \sim \text{Poisson}(\alpha)$ before sitting down to eat.



As more customers come in, each customer helps themselves to some of the buffet. The nth customer takes some of each dish k with probability $m_k n^{-1}$. After passing by every dish previous sampled, she then tries some of the next b many dishes with $b \sim \text{Poisson}(\alpha n^{-1})$. Worth noting is the fact that, like Dirichlet processes, this process imposes a "rich get richer" effect, as dishes that have been sampled many times are even more likely to be sampled again, corresponding to features which are highly expressed in a data sample.



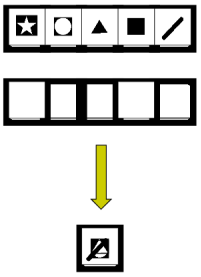
2 Extensions and Applications

2.1 A Linear Gaussian Model

Let us assume we have a set of images and a resulting image formed through a weighted combination of these images. In general, a latent factor model may be written in the form $X = WA^T + \epsilon$, where rows of A are latent features and rows of W are datapoint-specific weights for these features (ϵ is Gaussian noise). We can use the Indian Buffet Process to make an infinite factor model using the steps below[1].

- Sample $W \sim IBP(\alpha)$
- Sample $a_k \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$
- Sample $\epsilon_{nk} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2)$

[1]Not discussed in detail in class. See Griffiths and Ghahramani, 2006 for more details



2.2 Infinite Factor Analysis

The problem with the linear Gaussian model is that features are either selected or not (i.e. the Z matrix only has binary entries). We can make the model more sophisticated by adding weights to individual features. Let Z be the binary matrix generated by an Indian Buffet Process. These can be extended to a weights matrix W, by setting $W = Z \odot V$ where $\odot$ is an operator for the element-wise product of two matrices (called the Hadamard product). The matrix V now has the weights and can be generated, for example, from a normal distribution as $V \sim \mathcal{N}(0, \sigma_V^2)$. Now we can assume the following forms for the distributions of Z, V and A.

- $Z \sim IBP(\alpha)$
- $V \sim \mathcal{N}(0, \sigma_V^2)$
- $A \sim \mathcal{N}(0, \sigma_A^2)$

2.3 A Binary Model for Latent Networks

Consider the case where we have some observed binary data and we wish to learn their latent causes. A simple example is that of identifying biologically plausible 'latent' causes (illnesses) for a set of observed features (symptoms). This situation can be represented in terms of a *Noisy-OR* model (an alternative to the simpler Gaussian model described previously). Here rows and columns of Z represent symptoms and causes respectively. The likelihood model may be formulated as below.

- $Z \sim IBP(\alpha)$
- $y_{dk} \sim Bernoulli(p)$
- $P(x_{nd} = 1|Z, Y) = 1 - (1 - \lambda)^{z_n y_d^T}(1 - \epsilon)$

3 Inference Technique for IBP

The inference problem is that of inferring the latent variables (Z) given the observations (X). This is similar to inference in the Dirichlet process, where we wanted to find the indicator function and the prior centroids. In particular, we are interested in $P(Z_{nk}|Z_{-nk}, X, \theta)$[2] where n is a sample and k is a dimension. The IBP gives us a prior over Zs as $P(Z_{nk}|Z_{-nk}) \sim IBP$and given data likelihood, we can compute the posterior as

[2] θ here is the equivalent of the A referenced in the Gaussian Linear Models section

$P(Z_{nk}|Z_{-nk}, X, \theta) \propto P(Z_{nk}|Z_{-nk}, \alpha)P(X|Z, \theta)$. The first part is either m_k or $N - m_k$, depending on the values of Z_{nk} while the second part depends on the model.

3.1 Inference in the Restaurant Scheme

The exchangeability of rows and columns in IBP implies that we can treat any data point as if it was the last one. Let K_+ be the total number of used features, excluding the current data point. Let θ be the set of parameters associated with the likelihood. The prior probability of choosing one of these features is $\frac{m_k}{N}$. Using the formulae of the previous section, we can find the posterior probabilities as
$p(z_{nk} = 1|x_n, Z_{-nk}, \theta) \propto m_k f(x_n|z_{nk} = 1, Z_{-nk}, \theta)$
$p(z_{nk} = 0|x_n, Z_{-nk}, \theta) \propto (N - m_k) f(x_n|z_{nk} = 0, Z_{-nk}, \theta)$
In some cases we may be able to integrate θ out but otherwise it needs to be sampled. However, we are not done yet as we haven't addressed the probability of picking up a new feature. We can use the Metropolis Hastings method for this, which would essentially give us a probability which we would use to accept or reject a feature:

- Let K^*_{old} be the number of features appearing only in the current data point.
- Propose $K^*_{new} \sim Poisson(\frac{\alpha}{N})$ and let Z^* be the matrix with K^* new features appearing only in the current data point.
- With probability $\min(1, \frac{f(x_n|Z^*,\theta)}{f(x_n|Z,\theta)})$, accept the proposed matrix.

3.2 Inference using the Stick-Breaking Convention of Beta Processes

Consider the finite beta-Bernoulli model.
$\pi_k \sim Beta(\frac{\alpha}{K}, 1)$
$z_{nk} \sim Bernoulli(\pi_k)$
The z_{nk} are independent and identically distributed given the π_k values but are exchangeable if we integrate out the π_k. Under the limit of K tending to ∞, this can be used to obtain a distribution for IBP. This distribution over discrete measures is called the Beta process. Sample from the beta process have infinitely many atoms with masses between 0 and 1. Recalling that each atom of the beta process is the infinitesimal limit of a $Beta(\frac{\alpha}{K}, 1)$ random variable and that our observations for that atom are a $Binomial(\pi_k, N)$ random variable, the posterior is the infinitesimal limit of a $Beta(\frac{\alpha}{K} + m_k, N + 1 - m_k)$, since we know that beta distribution is a conjugate to the binomial. We can construct the beta process using a stick-breaking construction, analogous to the Dirichlet process stick-breaking construction. In this case, we sample a $Beta(\alpha, 1)$ random variable μ_k, break off a μ_k fraction of the stick, throw away what's left and recurse on the part of the stick left. Here π_k is a product of all the μ_j's for j from 1 to k (the atoms here do not sum up to 1). In order to use the stick-breaking construction for inferencing, we sample Z given π, θ and π given Z. The posterior for atoms for which m_k is positive is beta distributed while for those atoms for which $m_k = 0$, we can sample using the stick-breaking procedure. A slice sampler can be used if we do not wish to represent all atoms or a fixed truncation level.

4 Two Parameter Extension

The Indian Buffet Process, there is only a single parameter α that governs both the number of non-empty columns and the number of features per data point. To decouple these two, a two parameter extension is

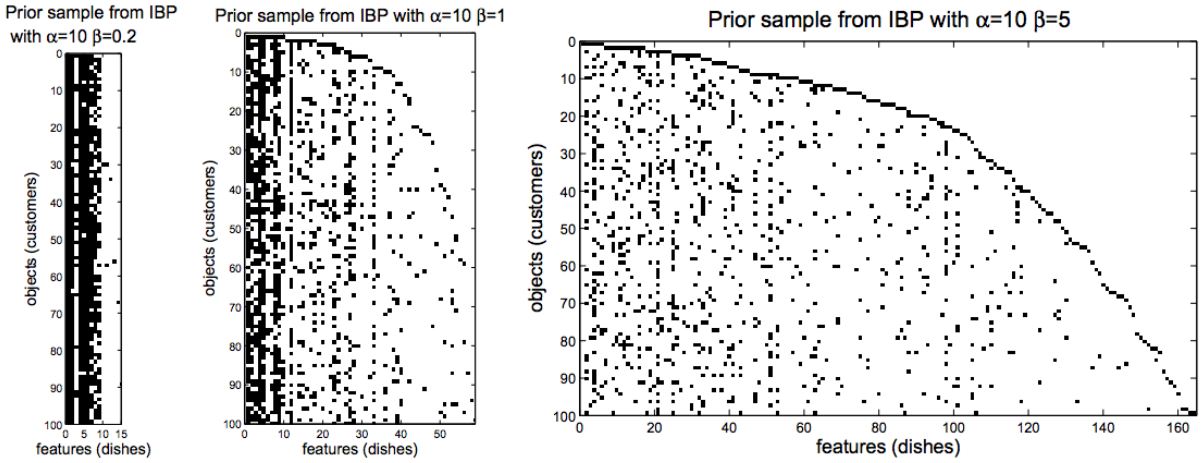


Figure 1: Three samples from two-parameter IBP with $\alpha = 10$ and $\beta = 0.2$(left),$\beta = 1$(middle) and $\beta = 5$(right)

proposed. To do this, the existing finite beta-bernoulli model is modified as follows:

$$\pi_k \sim Beta(\frac{\alpha\beta}{K}, \beta)$$

$$z_{nk} \sim Bernoulli(\pi_k)$$

The corresponding IBP is as follows:

1. First customer enters a restaurant with an infinitely large buffet.
2. He helps himself to $Poisson(\alpha)$ dishes.
3. The n^{th} customer enters the restaurant.
4. He helps himself to each dish with probability $\frac{m_k}{\beta+n-1}$
5. The n^{th} customer helps himself to each dish with probability $\frac{m_k}{\beta+n-1}$ where m_k is the number of times dish k was chosen. Here β behaves like imaginary customers before the first real customer comes in.
6. He tries $Poisson(\frac{\alpha\beta}{\beta+n-1})$ new dishes.

The number of features per data point is still marginally $Poisson(\alpha)$. The number of non-empty columns is now $Poisson(\alpha \sum_{n=1}^{N} \frac{\beta}{\beta+n-1})$. Please note that we recover the original IBP process by replacing $\beta = 1$. Figure 1 shows three matrices drawn from two-parameter IBP with $\alpha = 10$ but varying values of β. Although all three matrices have same number of non-zero entries, the number of features used varies considerably. At small values of β, features become shared by all objects. At higher values, features are more likely to be spread out.

Beta processes and IBP For IBP, we start with a Beta process - an infinite sequence of values between 0 and 1, that are distributed as the infinitesimal limit of the beta distribution. We combine this with a Bernoulli process to get a binary matrix. If the beta process is integrated out, we get an exchangeable distribution over binary matrices. This exchangeable distribution is similar to the one created by IBP.

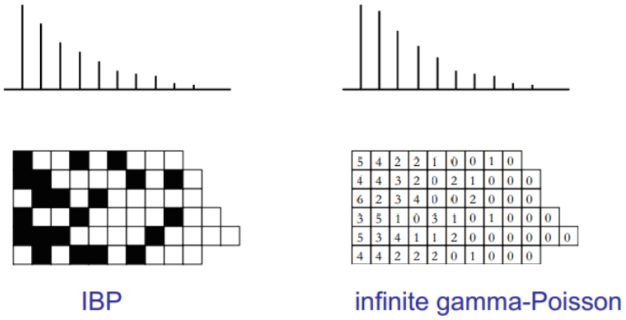


Figure 2: Difference between IBP and Gamma-Poisson process

5 Infinite Gamma-Poisson Process

The gamma process can be thought of as the infinitesimal limit of a sequence of gamma random variables. Or it can be viewed as follows:

$$D \sim DP(\alpha, H)$$

$$\gamma \sim Gamma(\alpha, 1)$$

$$G = \gamma D \sim GaP(\alpha H)$$

If we draw D from a Dirichlet process, and γ from a Gamma distribution, then the product γD comes from a gamma-Poisson process. We also know that gamma distribution is conjugate to Poisson process. We can associate each item v_k of the gamma process with a column of the matrix. We can generate entries for the matrix $z_{nk} \sim Poisson(v_k)$. This gives us entries as positive integers, as compared to binary numbers in the previous model. The use of this model can be in for instance, in unsupervised learning where multiple features are associated with the data and each feature can have multiple occurrences within each data point. Another use-case can be Bayesian matrix factorization, where a matrix of observations is decomposed into a product of two or more matrices with one of them being a non-negative integer valued matrix. In Fig 2, we show how different IBP and Gamma-Poisson process are. The histograms corresponds to sums over the columns in the matrices so sampled. And each row in the matrix corresponds to one observation, and column to a feature. For predicting distribution corresponding to the n^{th} row or the n^{th} customer, we can sample a count for every existing feature $z_{nk} \sim NegBinomial(m_k, \frac{n}{n+1})$. Sample $K_n \sim NegBinomial(\alpha, \frac{n}{n+1})$, and then partition this K_n according to Chinese Restaurant Process, and assign the resulting counts to the new columns.

6 Summary

The Indian Buffet Process is a generative model which may be used as a non-parametric prior for latent feature modeling. We saw how this model comes about as we consider the Beta-Bernoulli prior model as we take the number of latent features to go to the limit. Much as in a Dirichlet process we saw how the data dictate how some sparse subset of these latent features are leveraged in the model according to a "rich get richer" sampling strategy. We discussed the inference procedure for the Indian Buffet Process in the restaurant scheme and using the stick-breaking construction. Among some of the applications for the Indian Buffet Process, we went over the simple Linear Gaussian Model and the Binary Model for Latent Networks. To account for other things, like different modeling of number of non-zero features and non-binary counts, extensions were proposed. We discussed the two parameter extension and Gamma-Poisson processes.