10-708: Probabilistic Graphical Models 10-708, Spring 2014

21 : The Indian Buffet Process

Lecturer: Eric P. Xing *Scribes: Jesse Dodge and Nijith Jacob*

1 Motivation

1.1 Limitations of Latent Class Models

In latent class models, such as finite/infinite mixture models, we model a generative process in which the observed data points are realizations from some distribution determined by a *single* class. In the case of simple finite mixture models, the number of such classes has to be fixed a priori. This limitation is lifted in the case of infinite Dirichlet process mixture models by allowing an infinite number of latent classes. Even though such models can be used to perform clustering with a potentially infinite number of clusters, each point is limited to one cluster.

In several real world applications, data points could share multiple classes. For instance, consider the problem of clustering instances of human beings appearing in different contexts across a set of images. We want all such instances to share a cluster for human to different degrees depending on the context in which they appear. To model these problems where data points can share multiple clusters, latent feature models can be used.

2 Latent Feature Models

In Latent feature models, data points can exhibit multiple classes/features to varying degrees.

A very basic example of such a model is the factor analysis model given by

$$X = WA^T + \epsilon \tag{1}$$

where rows of A represent latent features, rows of W are the data point specific weights of these features and ϵ is gaussian noise.

Here the data X are modeled to be generated by a weighted combination of features. A is sometimes called the factor loading matrix and W, the coordinate matrix. One limitation of this model is that the number of features is finite and has to be specified a priori.

2.1 Infinite Latent Feature Models

In infinite latent feature models, we avoid pre-selecting the number of features. In the previous example of factor analysis, we want to make the A matrix to have an infinite column dimension and W to be a sparse weight matrix so that we can allow data to exist in an arbitrarily large feature space while avoiding the burden of explicitly representing A as an infinite feature matrix. This construction closely parallels

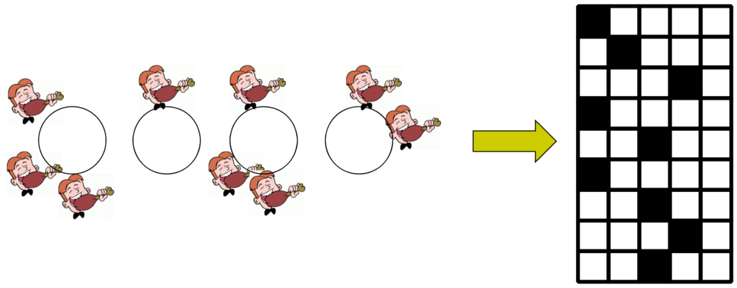


Figure 1: CRP as a distribution over binary matrices

the Dirichlet processes where we allow an infinite number of components while keeping the usage of these components to be purely based on the data.

To generate such infinite models, we need some way of constructing W, in so that it remains sparse with a potentially arbitrary number of non-zero columns. The Indian Buffet Process (IBP) is a process to generate such a weight matrix.

2.2 Indian Buffet Process

To see the rationale behind the Indian Buffet Process, we will first look at the Chinese Restaurant Process.

2.2.1 Chinese Restaurant Process: A distribution over binary matrices

The Chinese Restaurant Process (CRP) is used when we want to model an infinite number of clusters (tables). The trick used here is to not model a joint distribution over the cluster assignment, since to do so we have to know the number of tables in advance. Instead, we can look at each data point iteratively and use a predictive distribution to determine where a new data point is assigned conditioned on the previous data point assignments.

Figure 1 shows the binary matrix that results from the shown CRP assignments. Since each customer is assigned to a single table, each row in the binary matrix can only have one of the column to be set (black). In other words, the CRP allows every data point to use only one feature (table).

We can use a similar scheme to represent a distribution over binary matrices recording ”feature usages” across data, where each row corresponds to a data point and each column to a feature. In addition we want to allow every data point to use a small subset of features.

2.2.2 Indian Buffet Process Scheme

The Indian Buffet Process (IBP) is a process for generating distributions over binary matrices while allowing the data points to share multiple features. We sketch the algorithm below:

1. The first customer enters a restaurant with an infinitely large buffet.

2. He helps himself to the first $Poisson(\alpha)$ dishes

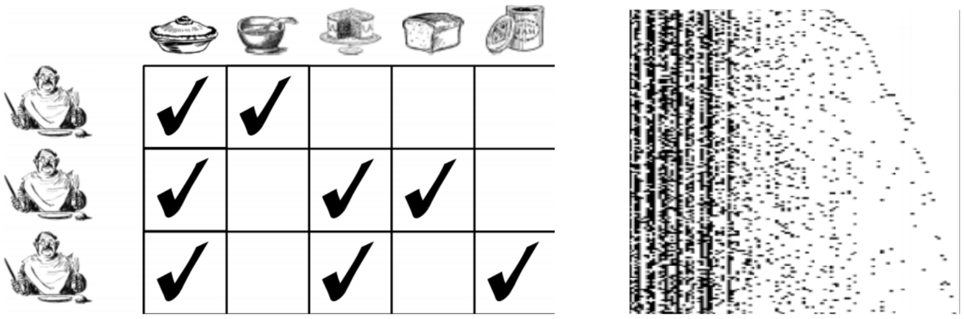


Figure 2: On the left is an example of Indian Buffet Process dish assignment for the first 3 customers. On the right is an example binary matrix generated from IBP.

3. The nth customer helps himself to each dish with probability m_k/n where m_k is the number of times dish k was chosen.

4. He tries $Poisson(\alpha/n)$ new dishes.

Figure 2 shown an example of Indian Buffet Process and an example of the binary matrix generated from it. One important feature, which you can see in the example, is that in the binary matrix, columns on the left hand side are denser (where the most-commonly selected tables are), and columns become more sparse as we move to the right.

2.2.3 Extending Binary Matrix to Weight Matrix

The Indian Buffet Process generates a binary indicator matrix, Z for each of the data points. These can be easily extended to a weight matrix, W, by setting $W = Z \odot V$ where $\odot$ indicate an element-wise (Hadamard) product of 2 matrices. The matrix V of weights, can be generated from some distribution, say a normal distribution, $V \sim \mathcal{N}(0, \sigma_V^2)$. Computing weight matrix in this way by decoupling the selection of features and the actual weights, generates a sparse weight matrix which avoids the problem of overfitting as in the case when a dense weight matrix is used.

2.2.4 Equivalence to infinite limit of Beta-Bernoulli Process

Consider the case of finite feature model with K known number of features. Let N be the number of data points. Now, place a Beta Bernoulli prior over Z, i.e.

$$\pi_k \sim Beta(\alpha/K, 1), k = 1, ..., K$$
$$z_{nk} \sim Bernoulli(\pi_k), n = 1, ..., N$$

It can be shown that the likelihood of Z generated by the Beta Bernoulli process in the limit that K tends to infinity is equivalent to the one generated by the Indian Buffet Process.

Since in the Beta Bernoulli process, we did not assume any ordering in the sampling of π_k and z_{nk}, the equivalence of IBP to this establishes that the ordering in IBP does not affect the likelihood.

2.2.5 Properties of IBP

These are some of the properties of the binary matrix generated by the Indian Buffet Process.

1. The number of dishes (features) chosen can arbitrary grow with the number of customers (data points).

2. The tendency to add new dishes decreases with customers as new dishes are added according to $Poisson(\alpha/n)$. As n increases, the Poisson distribution rate decreases.

3. On the other hand, the old dishes get reused based on the "rich get richer" principle.

4. The distribution generated by the Indian Buffet Process is *exchangeable*.

5. The number of non-zero entries in any row is distributed $Poisson(\alpha)$. We know that the first customer's row is distributed $Poisson(\alpha)$. From exchangeability, the average number of non-zero entries in other rows should also be distributed the same way.

6. The number of non-zero entries in the whole matrix is distributed $Poisson(N\alpha)$. This is true since we know that if $x_1 \sim Poisson(\alpha_1)$ and $x_2 \sim Poisson(\alpha_2)$, then $x_1 + x_2 \sim Poisson(\alpha_1 + \alpha_2)$

7. The number of non-empty columns is distributed $Poisson(\alpha H_n)$ where $H_n = \sum\limits_{n=1}^{N} \frac{1}{n}$

2.3 Two Parameter Extension

The Indian Buffet Process described earlier uses a single parameter α that determines the row occupancy and column occupancy. To decouple these two, we consider a two parameter extension. To do this, we incorporate an extra parameter β to the finite Beta Bernoulli process discussed earlier as given below.

$$\begin{aligned} \pi_k &\sim Beta(\alpha\beta/K, \beta), k = 1, ..., K \\ z_{nk} &\sim Bernoulli(\pi_k), n = 1, ..., N \end{aligned}$$

The corresponding two parameter IBP is as follows.

1. The first customer enters a restaurant with an infinitely large buffet.

2. He helps himself to the first $Poisson(\alpha)$ dishes

3. The nth customer helps himself to each dish with probability $m_k/(\beta + n - 1)$ 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.

4. He tries $Poisson(\alpha\beta/(\beta + n - 1))$ new dishes.

We see that the number of non-zero entries in a row are still distributed $Poisson(\alpha)$. But, the distribution over number of non-empty columns becomes $Poisson(\alpha \sum\limits_{n=1}^{N} \frac{\beta}{\beta+n-1})$.

Figure 3 shows the effect of the added parameter β. As β is increased, the matrix becomes more sparser while increasing the number of non-empty columns.

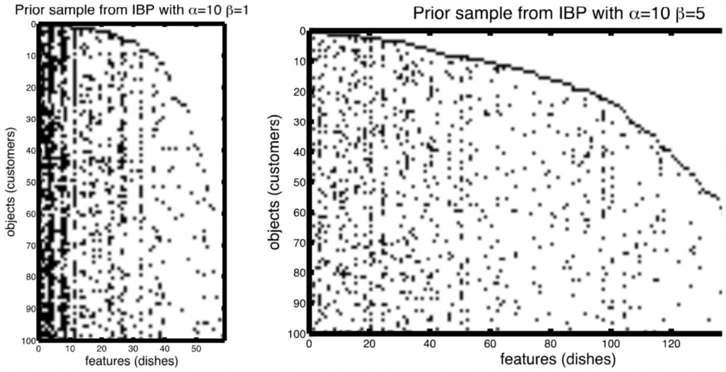


Figure 3: The two parameter extension to IBP. The left binary matrix is generated with $\alpha = 10, \beta = 1$ and the right with $\alpha = 10, \beta = 5$

2.4 Beta Processes and the IBP

The generative process described as IBP does not give any guarantees on convergence to a consistent density function. To do this, we have to make the connection of the process to some statistical measure.

Recall the relationship between Dirichlet processes and the Chinese Restaurant Process. Dirichlet process gives a prior over distribution of cluster weights. Integrating out these cluster weights gives us an exchangeable distribution over partitions of the data which is the same as the one generated by CRP.

In the case of IBP, we can establish a similar kind of relationship to a process called the Beta Process. The existence of such a measure is implied by the De Finetti's theorem which states that ”if a distribution $X_1, X_2,$ is exchangeable, there must exist a measure conditioned on which $X_1, X_2,$ are i.i.d.”

In the finite Beta Bernoulli model discussed earlier, we have that $\pi_k \sim Beta(\alpha\beta/K, \beta)$. The distribution over discrete measures obtained as the infinite limit of π_k as K tends to infinity is called the Beta process. Samples from the beta process have infinitely many atoms with masses between 0 and 1.

2.4.1 Posterior of Beta Process

Each atom of the Beta process is the infinitesimal limit of a $Beta(\alpha/K, 1)$ random variable. Our observation m_k for that atom is distributed $Binomial(\pi_k, N)$. Since beta is a conjugate for binomial distribution, the posterior is the infinitesimal limit of a $Beta(\alpha/K + m_k, N + 1 - m_k)$ random variable.

2.4.2 Stick Breaking construction for Beta Process

Paralleling the stick breaking construction for Dirichlet Process, there is a construction for Beta Process as well as follows.

1. Begin with a stick of unit length.
2. For $k = 1, 2, ...$
   (a) Sample a $Beta(\alpha, 1)$ random variable μ_k
   (b) Break off a fraction μ_k of the stick. This is the kth atom size.

(c) Throw away what's left of the stick.

(d) Recurse on the part of the stick that you broke off

π_k is then given by $\pi_k = \prod_{j=1}^{k} \mu_j$. Note here that the π_k here does not not sum to 1 unlike the stick breaking construction in Dirichlet Process. The π_ks generated can then be used to determine how often each feature is used for the data points.

2.5 Binary Model for latent networks

If we're interested in discovering latent causes for observed binary data, such as discovering the diseases (latent causes) a medical patient has just from observing their symptoms (observed binary data), we can use an IBP. Specifically, we can formulate a likelihood model for the IBP in terms of a Noisy-Or model, as follows:

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

The intuition of this formulation is that each y_{dk} is an indicator feature for symptom k of illness d. This formulation is not unique when trying to solve the patient-illness problem presented above, however. An alternative to the proposed solution would be to use a multinomial distribution over the symptoms, instead of a number of bernoulli distributions, for example.
Now that we have some likelihood models, we can perform inference.

3 Inference

The inference problem: Given X, can we find Z (and maybe A)? (See Forumla 1.) This is similar to the Dirichlet process, where we wanted to find the indicator function and also the prior centroids. Specifically, we're interested in

$$P(Z_{nj}|Z_{-nj}, X, A)$$
$$P(A|Z, X)$$

where n indicates a sample and j indicates a particular dimension. The IBP gives us the prior over the Zs, specifically $IBP \sim P(Z_{n,j}|Z_{-n,-j})$, and now that we have data likelihood (from the previous section), we can compute the posterior.

$$P(Z_{nj}|Z_{-nj}, X, A) \propto P(Z_{nj}|Z_{-nj})P(X|Z_{nj}, A)$$

3.1 In the restaurant scheme

When using the restaurant construction of the IBP, our analysis relies on a few points. First, that exchangability means we can treat each data point as our last. Second, the prior probability of choosing a feature is

m_k/N. If we assign Θ to be the set of parameters for the likelihood, then we have:

$$P(z_{nk} = 1|x_n, Z_{-nk}, \Theta) \propto m_k f(x_n|z_{nk} = 1, Z_{-nk}, \Theta)$$
$$P(z_{nk} = 0|x_n, Z_{-nk}, \Theta) \propto (N - m_k) f(x_n|z_{nk} = 1, Z_{-nk}, \Theta)$$

where $P(z_{nk} = 1|x_n, Z_{-nk}, \Theta)$ is the probability that we will choose an existing feature, $P(z_{nk} = 0|x_n, Z_{-nk}, \Theta)$ is the probability we will choose a new feature, and $f(x_n|z_{nk} = 1, Z_{-nk}, \Theta)$ is the marginal likelihood of the data.
Using this construction, we can run MCMC. You can do a Metropolis-Hastings step to propose a new feature as necessary.

3.2 In the stick-breaking construction

When using the stick-breaking construction of the IBP, we can sample from $P(Z|\Pi, \Theta)$ and $P(\Pi|Z)$. In this setup, the following is true:

1. The posterior for atoms for which $m_k > 0$ are Beta distributed.
2. The atoms for which $m_k = 0$ can be sampled using the stick-breaking procedure
3. Slice-sampling can be used to avoid representing all the atoms.

4 Other distributions over infinite, exchangeable matrices

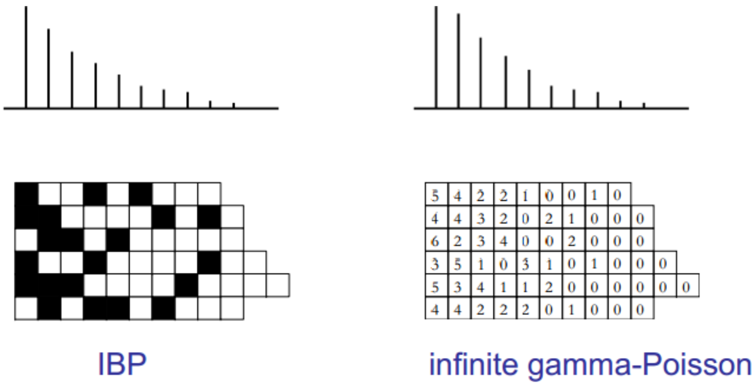
Within the IBP, the Z matrix is used for feature selection. One can imagine other uses for this kind of approach, however. For example, one can represent an infinite network in this matrix. However, one must be careful with this approach is in an adjacency matrix the rows and columns encode the same information. These matrices of infinite row and column size are useful for a number of things. For example,

4.1 The Infinite Gamma-Poisson Process

Recall that the Beta process can be described as the infinitesimal limit of a sequence of beta random variables. Similarly, instead of using beta random random variables, we can use gamma random variables. This leads to the gamma-Poisson process.

$$\text{if } D \sim DP(\alpha, H)$$
$$\text{and } \gamma \sim Gamma(\alpha, 1)$$
$$\text{then } G = \gamma D \sim GaP(\alpha, D)$$

Here if we draw D from a Dirichlet process, then γ from a Gamma distribution, then the product γD comes from a gamma-Poisson process. This ends up quite similar to the IBP. With the IBP, for a given observation, we take draws from bernoulli distributions to see if a given feature is used. However, with the gamma-Poisson process, we can generate the z_{nk} with a poisson distribution whose rate is defined by the gamma process. This gives us a matrix of positive integers, as opposed to a binary matrix of indicator values. See Figure 4. When predicting the nth row of the matrix, we can sample a count $z_{nk} \sim NegativeBinomial(m_k, n/(n+1))$. This gives the values in the matrix the "rich get richer" property, which tells us that if $\sum_{n\prime \in 1 \ldots n-1} z_{n'k}$ is high, then z_{nk} will be high with a high probability.



5	4	2	2	1	0	0	1	0			
4	4	3	2	0	2	1	0	0	0		
6	2	3	4	0	0	2	0	0	0		
3	5	1	0	3	1	0	1	0	0	0	
5	3	4	1	1	2	0	0	0	0	0	0
4	4	2	2	2	0	1	0	0	0		

Figure 4: The IBP compared to the GaP. The histograms above correspond to sums over the columns in the matrices below. In the matrices, each row corresponds to one observation, and each column corresponds to one feature.

While applications for this process may be difficult to find, constructing these processes is intellectually interesting. Building these conjugate pairs (such as Dirichlet to Dirichlet process, gamma to gamma-bernoulli process, beta to beta-bernoulli process, and gamma to gamma-poisson process) is interesting, as these have nice properties that allow for sampling that gives a correct posterior distribution.

5 Summary

Modeling assumptions typically fall somewhere between two extremes. On one side, nonparametric approaches make no assumptions about the models that generated the data, while Bayesian approaches typically make parametric assumptions about the class of models for the data as well as having priors over the parameters to those models. The approaches covered in this lecture, nonparametric Bayseian models, fall somewhere in between. Since we don't know, a priori, how many features we will be using, but we do have a prior over the number of features used, we have advantages from both extremes.