10-708: Probabilistic Graphical Models, Spring 2020

24: Indian Buffet Process

Lecturer: Eric P. Xing *Scribes: Stefan Andjelkovic, David Bick, Prithvi Gudapati*

1 Recap

1.1 Motivating Examples

To understand why we would be interested in these techniques, it is best to start with a motivating example. We see concretely how we could apply the method, and can keep it in mind when observing the technical details.

Imagine we have four ground truth features that we can combine in images, shown below

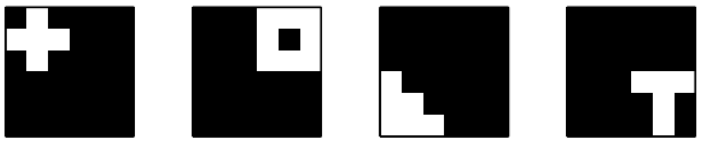

Figure 1: Ground truth features

and we observe combinations of these features that have been corrupted with noise, as shown below

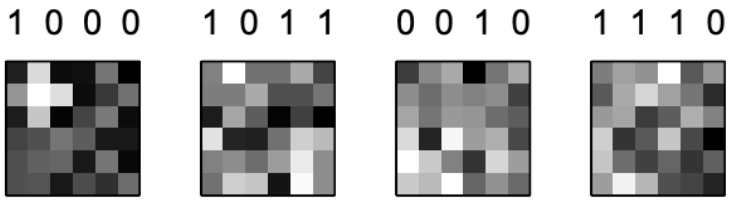

Figure 2: Noisy Combinations of Features

where the 1's and 0's above the image show which of the four latent features are present. In reality, we would observe the corrupted images and not have any idea what the ground truth features are. We would

not even know how many features to expect that there would be. This is the motivation behind the Indian Buffet Process. We would start with the second set of images, and produce the following

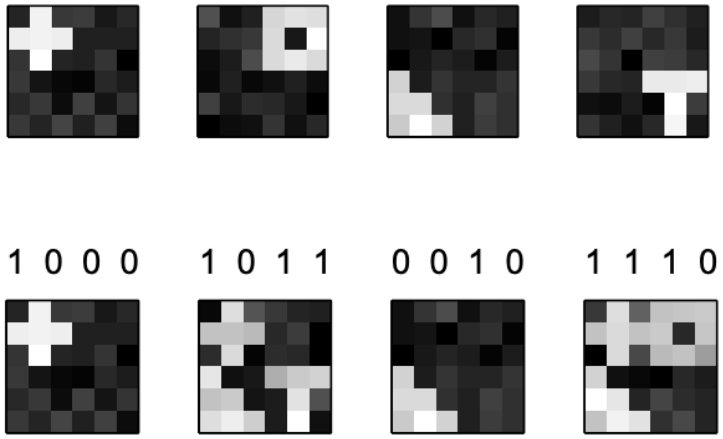

Figure 3: IBP Outputs

where the first set of images is the posterior mean for the latent features that our images are comprised of, and the second set of images is the set of estimates of which features the observed data are composed of. Thus the IBP learned both how many latent features there are in the data, and what these features look like. It also gives an estimate of which features are present in a given observation, the vector above the second set of images.

The images and much of the information in these notes is taken from Griffith and Ghahramani [1].

1.2 General Bayesian Nonparametrics

The general motivation for Bayesian Nonparametrics is to allow an arbitrary number of elements for a desired quantity. In the above example, the finite version is a fixed latent feature model, which was developed before the infinite version. Another classical example is mixture model clustering, where before the Dirichlet process mixture model was developed, one had to specify a number of clusters and iterate over different quantities for this value to see a metric showed diminishing returns.

The bayesian nonparametrics key contribution is allowed the model to add clusters (in the Dirichlet mixture model) or latent variables (IBP) as needed according to the data. To do this, you must specify an unbounded prior distribution on clusters or latent variables. This is generally referred to as an infinite distribution, however once you have finite observed data, which is always the case, it reduces the infinite distribution is limited to a finite instantiation.

2 Latent variable models

We assume that we have a collection of features $\mathbf{F} = [f_1^T f_2^T \dots f_N^T]^T$ [1]. We are concerned then with both $P(X|F)$ and P(F), where $P(X|F)$ is the likelihood for each observation given its set of features, and P(F) is the prior over different matrices of features. The difference between finite and infinite latent variable models comes from P(F), and whether it places a bound on the number of variables.

We decompose $F = Z \otimes V$, where Z is a binary matrix where $z_{ik} = 1$ if object i has feature k, and 0 otherwise [1]. We use $\otimes$ to represent element-wise product, also called Hadamard product. V represents the values that the feature takes if present, and Z determines the presence in F. We can thus decompose p(F) = p(F)p(V).

We break F into the $Z \otimes V$ because it simplifies the problem to finding a infinite prior on binary matrices. Then we can worry about the other values of the features, V, separately. The binary matrices are useful in many applications and so reducing the scope to Z does not reduce the relevance of the study. Also, and most importantly, (finite) binary matrices can be specified by a Beta-Bernoulli process, which has closed form due to the conjugacy of the Beta prior with the Bernoulli likelihood. This allows us to take the limit of the closed form integral, which is not generally possible for any combination of likelihood and prior.

2.1 Finite Variant

We can define the probability of a binary matrix as follows:

$$P(Z|\pi) = \Pi_{k=1}^{K} \Pi_{i=1}^{N} P(z_{ik}|\pi_k) \tag{1}$$

$$= \Pi_{k=1}^{K} \pi_k^{m_k} (1 - \pi_k)^{N - m_k} \tag{2}$$

where π_k is the probability of an object containing the i-th feature, and m_k is the number of objects that contain feature k.

The conjugate prior for this likelihood is the beta distribution as a prior on each π_k. We can isolate $P(Z)$, rather than $P(Z|\pi)$, by multiplying by the prior to get the joint $P(Z, \pi)$ and integrating out π. This is shown below, with a $Beta(\frac{\alpha}{K}, 1)$ prior (see [1] for details on the parameters of this prior):

$$P(Z) = \Pi_{k=1}^{K} \int (\Pi_{i=1}^{N} P(z_{ik}|\pi_k)) p(\pi_k) d\pi_k \tag{3}$$

$$= \Pi_{k=1}^{K} \frac{B(m_k + \frac{\alpha}{K}, N - m_k + 1)}{B(\frac{\alpha}{K}, 1)} \tag{4}$$

$$= \Pi_{k=1}^{K} \frac{\frac{\alpha}{K} \Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \tag{5}$$

2.2 Infinite Limit

To define the infinite limit of eq (5), we must note briefly that we actually consider the equivalence class of binary matrices, not individual binary matrices. The essence of the idea is that if we reorder the columns, we have not actually changed anything about the matrix because the columns are independent. We denote the equivalence class of matrices as $[Z]$ (see [1] for more details).

We break equation (5) into two components, where we denote K_0 as the number of features with $m_k = 0$, and K_+ as the number of features with $m_k > 0$. The result of the decomposition is (7), where (5) is repeated for convenience,

$$\Pi_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \tag{6}$$

$$= (\frac{N!}{\Pi_{j=1}^{N}(j + \frac{\alpha}{K})})^K (\frac{\alpha}{K})^{K_+} (\Pi_{k=1}^{K} \frac{(N - m_k)!\Pi_{j=1}^{m_k - 1}(j + \frac{\alpha}{K})}{N!}) \tag{7}$$

We then take the limit of equation (7) to get the infinite latent variable model

$$\lim_{K\to\infty} (\frac{N!}{\Pi_{j=1}^{N}(j + \frac{\alpha}{K})})^K (\frac{\alpha}{K})^{K_+} (\Pi_{k=1}^{K} \frac{(N - m_k)!\Pi_{j=1}^{m_k - 1}(j + \frac{\alpha}{K})}{N!}) \tag{8}$$

$$= (\frac{\alpha^{K_+}}{\Pi_{h=1}^{2^N - 1} K_h!}) \exp(-\alpha H_N) \Pi_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!} \tag{9}$$

The main takeaway is that we have used the closed form of the Beta-Bernoulli model that was facilitated by conjugacy, and taken the limit as number of columns goes to infinity, which was also facilitated by the the equivalence classes to simplify the number of binary matrices. We end with a result that only depends on K_+, so our infinite model depends only on what we observed in our finite dataset, although we allowed an infinite number of features a priori. There is an added term that we get by taking $P([Z])$ rather than $P(Z)$, which counts the number of matrices per equivalence class, see [1] for further details.

3 Predictive distribution: Indian Buffet Process

3.1 The Indian Buffet Process

We can describe a model using an analogy to an Indian restaurant with an infinitely large buffet. In this process, each customer besides the first customer chooses dishes partially based on the customers who came before them. It begins with the first customer helping himself to Poisson(α) dishes. Each subsequent customer helps himself to each of the previously chosen dishes with a probability of $\frac{m_k}{n}$, where m_k is the number of customers before him who took kth dish and n is the number of customers so far including him. Once he has taken dishes that have already been taken by other customers, he proceeds to try Poisson($\frac{\alpha}{n}$) new dishes. Figure 4 is an illustration of what this process would look like with the columns representing the different dishes and the rows representing the customers.

4 Properties of the Indian Buffet Process

The Indian Buffet Process has a rich getting richer property as popular dishes become more popular. This is due to the fact that the probability that a dish gets picked goes up when it is popular. We can see this effect in Figure 5. Once the first few customers pick dishes on the left side, those dishes continue to get picked. However, there continues to be some exploration.

In addition, the number of nonzero entries for each row is distributed according to Poisson(α) due to exchangeability, which means that changing the ordering of the rows and columns in the matrix does not change the joint probability. This allows any row or column to be treated as the last row or column in the matrix.

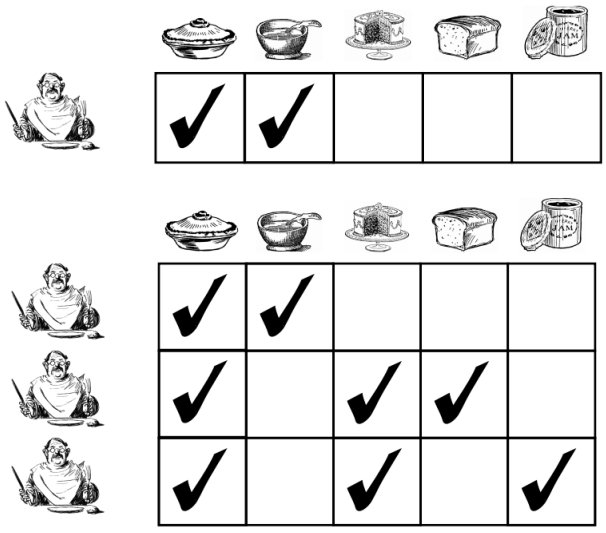

Figure 4: IBP Example

Furthermore, given that if $x_1 \sim \text{Poisson}(\alpha_1)$ and $x_2 \sim \text{Poisson}(\alpha_2)$ then $(x_1 + x_2) \sim \text{Poisson}(\alpha_1 + \alpha_2)$, the number of nonzero entries for the whole matrix is distributed according to Poisson($N\alpha$). Furthermore, the number of non-empty columns is distributed according to Poisson($\alpha H N$)

4.1 Relation to Infinite Beta-Bernoulli Model

Now the goal is to show that the Indian Buffet Process is lof-equivalent to the infinite beta Bernoulli model described earlier.

The probability of a matrix Z generated using this process can be computed to be

$$p(\mathbf{Z}) = \prod_{n=1}^{N} p(\mathbf{z}_n | \mathbf{z}_{1:(n-1)}) \tag{10}$$

$$= \prod_{n=1}^{N} \text{Poisson}\left(K_1^{(n)}\right) \prod_{n-1}^{K_+} \left(\frac{\sum_{i=1}^{n-1} z_{ik}}{n}\right)^{z_{nk}} \left(\frac{n - \sum_{i=1}^{n-1} z_{ik}}{n}\right)^{1-z_{nk}} \tag{11}$$

$$= \frac{\alpha^{K_+}}{\prod_{n=1}^{N} K_1^{(n)}!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!} \tag{12}$$

If we include the cardinality of $\mathbf{Z}$, we can see that this is the same as Equation (9). Thus, we can see that that matrix can be sampled using the Indian Buffet Process.

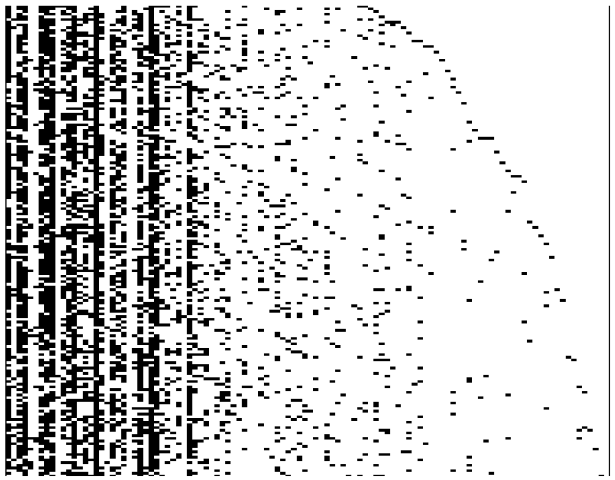

Figure 5: Indian Buffet Process Left Ordering Bias

5 Building latent feature models using the IBP

Now the goal is to use the Indian Buffet Process to build latent feature models.

A simple model that can be developed using the Indian Buffet Process is the linear Gaussian model.

The general form of any latent factor model is

$$\mathbf{X} = \mathbf{W}\mathbf{A}^T + \epsilon, \tag{13}$$

where $\mathbf{W} = \mathbf{Z} \odot \mathbf{V}$. For a linear Gaussian model, $\mathbf{W} = \mathbf{Z}$, so $\mathbf{W}$ is just a binary matrix. $\mathbf{Z}$ is sampled using the Indian Buffet Process, meanwhile $\mathbf{a}_k \sim \mathcal{N}(\mathbf{0}, \sigma_a^2\mathbf{I})$ and $\epsilon_{nk} \sim \mathcal{N}(0, \sigma_\epsilon^2)$

The linear Gaussian model is quite constrained as it is an all-or-nothing model due to the binary "loading matrix" W.

As a result, it is better to not set $\mathbf{W} = \mathbf{Z}$, and instead, make $\mathbf{W}$ a weight matrix. This can be done by making $\mathbf{V}$ a weight matrix that is created using some distribution, for example a Gaussian.

6 Inference in the IBP

Exchangeability, which was mentioned previously as an important property for the Indian Buffet Process, plays an important role in inference.

With K_+ being the total number of used features, excluding the current data point, and Θ being the set of parameters associated with the likelihood, the prior probability of choosing one of these features if $\frac{m_k}{N}$. This is because we can treat the current data point as the last one using exchangeability.

In addition, the posterior probability is proportional to

$$p(z_{nk} = 1|\mathbf{x}_n, \mathbf{Z}_{-nk}, \Theta) \propto m_k f(\mathbf{x}_n|z_{nk} = 1, \mathbf{Z}_{-nk}, \Theta) \tag{14}$$
$$p(z_{nk} = 0|\mathbf{x}_n, \mathbf{Z}_{-nk}, \Theta) \propto (N - m_k) f(\mathbf{x}_n|z_{nk} = 0, \mathbf{Z}_{-nk}, \Theta), \tag{15}$$

where f is the likelihood of the sample given how the features are chosen.

Now, new features must be added as well. This can be done using the Metropolis Hastings method.

1. Propose $K^*_{new} \sim \text{Poisson}(\frac{\alpha}{N})$, and let Z^* be the matrix with K^*_{new} features appearing only in the current data point.
2. Accept the proposed matrix with probability

$$\min\left(1, \frac{f(\mathbf{x}_n|Z^*, \Theta)}{f(\mathbf{x}_n|Z, \Theta)}\right)$$

7 Beta processes and the IBP

In the finite Beta-Bernoulli process, we had $\pi_k \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right)$ and $z_{nk} \sim \text{Bernoulli}\left(\pi_k\right)$. Integrating π_k out leaves exchangeable z_{nk}. We defined Indian Buffet Process as infinite limit of the beta random variables, when $K \to \infty$.

Beta process was defined by Hjort [2], and more on its extension for IBP can be found here [3]. In the scope of this topic, and for the sake of simplicity, we can define beta process as a process characterized by distribution over discrete measures, like the one shown above.

Posterior distribution of the column probabilities can be obtained in the closed form. Beta process atoms (π_k) are drawn from beta distribution, and their counts from Binomial(π_k, N). Because beta distribution is conjugate to binomial, the posterior for each k is Beta $\left(\frac{\alpha}{K} + m_k, N + 1 - m_k\right)$.

To combine all the samples into joint posterior, we can use the beta process stick-breaking construction, as presented in [4].

7.1 Stick-breaking construction

The stick-breaking construction for the beta process is as follows:

- Start with a unit length stick and define $\pi_0 = 1$.
- For each k, sample $\mu_k \sim \text{Beta}(\alpha, 1)$.
- Break off a stick part of length μ_k and throw away the rest.
- Store the value $\pi_k = \pi_{k-1}\mu_k$.

This process is shown in Fig. (6)

Unlike Dirichlet process, in beta process atom values do not necessarily sum up to 1.

Inference in beta process can be performed by sampling distributions $\mathbf{Z} \mid \pi, \mathbf{\Theta}$ and $\pi \mid \mathbf{Z}$. Posteriors for atoms with $m_k > 0$ are beta distributed and posteriors for atoms with $m_k = 0$ can be sampled with stick-breaking procedure.

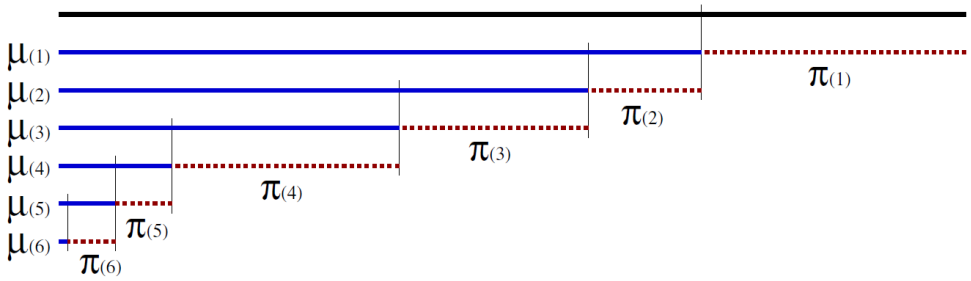

Figure 6: Stick-breaking construction for beta processes, adapted from [4]. Note that red dotted lines correspond to Dirichlet process π_k values, while blue solid lines correspond to beta process μ_k values

7.2 Two-parameter extension

In the previous examples, parameter α determines both *the number of non-empty columns* and *the number of features per data point*. These wouldn't necessarily be equal, so they could be decoupled. The proposed solution is to, instead of sampling $\pi_k \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right)$, sample weights from:

$$\pi_k \sim B\left(\frac{\alpha\beta}{K}, \beta\right)$$

In this extension restaurant scheme is as follows:

- A customer walks into the restaurant and orders Poisson(α) dishes.
- The n-th customer walks into the restaurant and orders previous dishes, each with probability $\frac{m_k}{\beta+n-1}$ of being ordered.
- Then, he orders Poisson($\frac{\alpha\beta}{\beta+n-1}$) new dishes.

This procedure guarantees that marginal number of features follows $N_{\text{features}} \sim \text{Poisson}(\alpha)$, and number of non-empty columns:

$$N_{\text{non-empty columns}} \sim Poisson\left(\alpha \sum_{n=1}^{N} \frac{\beta}{\beta+n-1}\right)$$

When $\beta = 1$, we recover Indian Buffet Process.

8 The infinite gamma-Poisson process

In analogy to the IBP as infinitesimal limit of beta-Bernoulli process, we can define gamma process as infinitesimal limit of gamma random variables, drawn from Poisson distribution. If D is Dirichlet process, drawn from $D \sim \text{DP}(\alpha, H)$, and γ from gamma distribution $D \sim \text{Gamma}(\alpha, 1)$, then gamma process $G = \gamma D$ is distributed according to the $G \sim \text{GaP}(\alpha, H)$.

Gamma distribution is conjugate to the Poisson distribution, so for each atom v_k of gamma process we can sample $z_{nk} \sim \text{Poisson}(v_k)$

Alternatively, to construct matrix $\mathbf{Z}$, row by row, we can modify IBP. For each row n:

- For each of the previous features sample a count $z_{nk} \sim \text{NegBinom}\left(m_k, \frac{n}{n+1}\right)$.
- Sample total count of new features $K_n^* \sim \text{NegBinom}\left(\alpha, \frac{n}{n+1}\right)$.
- Partition K_n^* according to the Chinese Restaurant Process, and assign these counts to the new features.

More on the infinite gamma-Poisson process and how it applies to computer vision can be found in [5].

References

[1] Thomas L. Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *J. Mach. Learn. Res.*, 12:1185–1224, July 2011.

[2] Nils Lid Hjort et al. Nonparametric bayes estimators based on beta processes in models for life history data. *the Annals of Statistics*, 18(3):1259–1294, 1990.

[3] Romain Thibaux and Michael I Jordan. Hierarchical beta processes and the indian buffet process. In *Artificial Intelligence and Statistics*, pages 564–571, 2007.

[4] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *Artificial Intelligence and Statistics*, pages 556–563, 2007.

[5] Michalis K Titsias. The infinite gamma-poisson feature model. In *Advances in Neural Information Processing Systems*, pages 1513–1520, 2008.