

## The Indian Buffet Process: An Introduction and Review

Lecturer: *Alternative Scribe*Scribe: *Genevieve Flaspohler*

## 1 Background

This background section was adapted from the introduction in Blei et al. (2010).

There are two major types of machine learning problems: supervised and unsupervised. In a supervised problem, we are provided with data and an associated label for each data point. The crux of these algorithms is to find a suitably robust and accurate function that maps from our data space  $\mathcal{X}$  to our label space  $\mathcal{Y}$ ,  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . Unsupervised learning tackles a fundamentally different, and perhaps more challenging, problem. Given only the data, unsupervised learning seeks to infer useful structure or patterns within this data. Classical unsupervised learning problems include clustering, density estimation, and dimension reduction.

In supervised learning problems, the data labels provide the problem with enough structure to learn a meaningful function  $f$ . In unsupervised learning, this structure is not inherent in the data; the problem designer must provide structure in the form of a probabilistic generative model. These generative models define what “structure” or “pattern” mean in the context of the data. Generative models are often specified in the form of a probabilistic graphical model, allowing inference and belief updates to take place within a fully Bayesian framework.

In this class, we learned several unsupervised learning models and their generative processes. For example, the Latent Dirichlet Allocation (LDA) specifies a generative process where data are modeled as a collection of documents, each of which is generated from a mixture of global topics (Fig. 1). Bayesian inference in this model is tractable because of the conditional conjugacy relationships embedded in the graphical model.

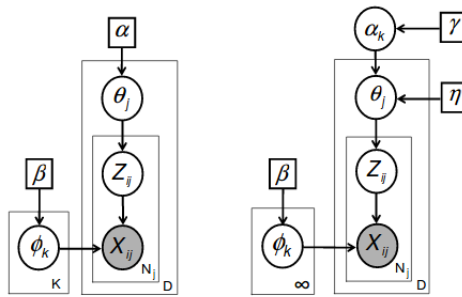


Figure 1: The graphical model representing the generative process for LDA (left) and HDP (right) [Newman et al. (2009)]. The shaded nodes represent the observed data, and the rest of the structure in the problem is provided by the graphical model.

LDA is an example of a parametric unsupervised learning model; the number of global topics present in the data is pre-specified and cannot grow to accommodate additional data. Since the majority of the structure in unsupervised learning problems is imposed by the generative model, a generative model with strong,

parametric assumptions will tend to inform the discovered structure more than the data itself. Additionally, many data have a fundamentally growing number of parameters: the more documents you analyze, the more topics you are likely to discover; the more animals you find on earth, the more species you will discover. These data could not be accommodated by a parametric model.

Bayesian nonparametric (BNP) inference seeks to overcome these limitations. BNP models replace parametric priors and posteriors with stochastic processes. Stochastic processes are infinite, indexed collections of random variables. By allowing the number of parameters to be unbounded and grow with the data, BNP models provide the weakest assumptions possible, allowing the “data to speak”, while still providing enough structure to learning meaningful patterns in the data. In this class, we have seen several BNP models. For example, the HDP topic model, a nonparametric extension of LDA, employs a nested Dirichlet process to allow the number of global topics within the corpus to be determined by the data (Fig. 1).

## 2 Motivation

In these notes, we will explore the problem of data clustering and cluster property inference. For example, given data generated from a mixture of three Gaussian distributions, we would wish to infer the cluster membership of each data point and the mean and variance of each cluster. A Dirichlet process (DP) and its marginal, the Chinese Restaurant Process (CRP) provide an elegant nonparametric solution to this cluster problem (Fig. 2).

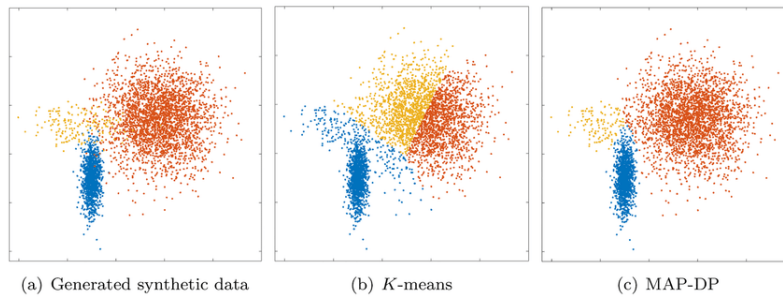


Figure 2: Given simulated 2- $D$  data (a), a comparison between the clusters recovered by a parametric  $K$ -means clustering algorithm (b) and nonparametric DP clustering (c) [Raykov et al. (2016)]

As we have seen in Section 1, the structure imposed on an unsupervised learning problem, via a generative process, plays a critical role in the latent structure discovered in the data. Given identical data, different generative processes will infer different latent structures. For our clustering problem, the DP structure models each data point as being generated from only one cluster and its observable properties are determined solely by this cluster identify. For some data, this assumption is too limiting. The Beta process, and its marginal the Indian Buffet Process (IBP), introduces a more flexible generative model. Under the IBP, each data point can belong to many different classes (i.e. features) and the point’s final observable properties are determined by a combination of the influences of all of its features.

The difference between the structure imposed by the CRP and the IBP is highlighted in Figure 3. In the CRP, a data point (customer) is identified with a single cluster (table) and the data point’s properties are entirely determined by that cluster. In the IBP, each data point (customer) can be associated with a potentially infinite number of dishes (features), which influence the data point’s expression. We will explore the formulation of the IBP generative process and its implications on the discovered latent structure in data in the rest of the notes.

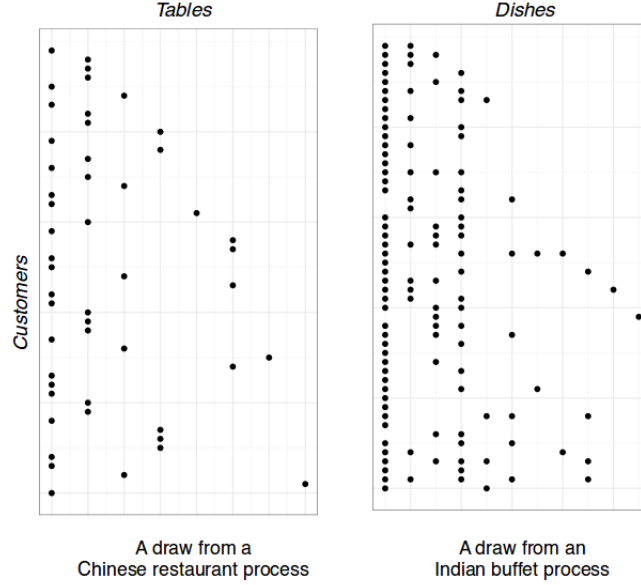


Figure 3: A comparison between the structure implied by the CRP model (left) vs the IBP model (right). In the CRP, a data point (customer) is identified with a single cluster (table) and the data point’s properties are entirely determined by that cluster. In the IBP, each data point (customer) can be associated with a potentially infinite number of dishes (features), which influence the data point’s expression. The above matrices represent this structure visually [Gershman and Blei (2011)].

### 3 Indian Buffet Process Model

In Griffiths et al. (2011), the authors define the Indian Buffet Process (IBP) as: ”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.” I will expand upon this definition in the following sections, keeping in mind the right matrix in Figure 3.

#### 3.1 Latent Class Models

Taking a step back for a moment to our classical clustering problem, we have  $N$  objects. The  $i$ th object has  $D$  observable properties represented by a row vector  $\mathbf{x}_i$ . Our  $N \times D$  data matrix  $\mathbf{X} = [\mathbf{x}_1^T \mathbf{x}_2^T \dots \mathbf{x}_N^T]$  indicates the properties of all  $N$  objects. In a mixture model, where each data point can only belong to one class, we would also define a class vector  $\mathbf{c} = [c_1 c_2 \dots c_N]^T$  to indicate the class membership of each data point. A mixture model is specified by a prior over class assignments  $P(\mathbf{c})$  and the likelihood of our data given the class assignments  $P(\mathbf{X}|\mathbf{c})$ . In the IBP model, we will modify both the prior and the likelihood function to allow each data point to be associated with an unbounded number of classes.

#### 3.2 Latent Feature Models

In a latent feature model, we again have  $N$  objects, with the  $i$ th object having  $D$  observable properties represented by a row vector  $\mathbf{x}_i$ . However, we now assume that these properties are generated from a

distribution determined by a latent feature vector  $\mathbf{f}_i$  associated with each data point. Instead of a class vector, we now have a feature matrix  $\mathbf{F} = [\mathbf{f}_1^T \mathbf{f}_2^T \dots \mathbf{f}_N^T]$ . Our model is specified with the corresponding prior over features  $P(\mathbf{F})$  and the likelihood of our observed properties given the features,  $P(\mathbf{X}|\mathbf{F})$ .

In this work, the prior over features  $P(\mathbf{F})$  is broken into two component matrices: a binary matrix  $\mathbf{Z}$  that indicates if a feature is possessed by each data point, and a matrix  $\mathbf{V}$  which indicates the value of the feature for each data point. This decomposition is visualized in Figure 4. The paper focuses on defining a prior over  $\mathbf{Z}$ , since knowing  $\mathbf{Z}$  is sufficient to determine the dimension of  $\mathbf{V}$  and therefore  $\mathbf{F}$ .

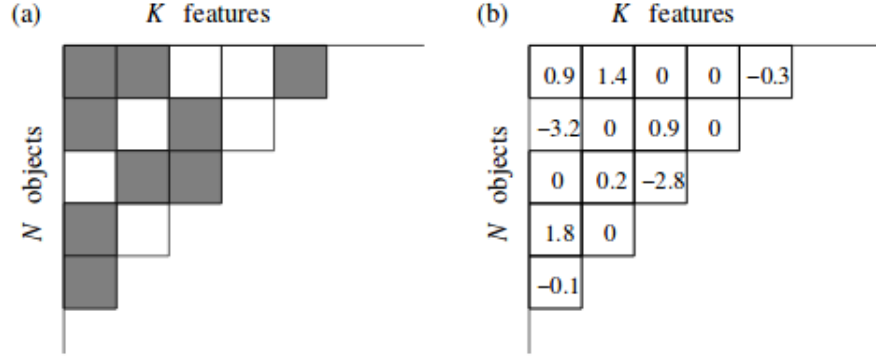


Figure 4: The decomposition of the feature vector  $\mathbf{F}$  into a binary, indicator matrix  $\mathbf{Z}$  (left) and a value matrix  $\mathbf{V}$  (right). The value matrix can be either real or integer valued.

### 3.3 Defining a prior over infinite binary matrices

In our previous nonparametric clustering problem, we defined a prior over an unbounded number of clusters in a variety of ways (e.g. as the limit of a finite model or via a simple stochastic process - the CRP). We can also define our desired prior  $P(\mathbf{Z})$  in these two ways, as we will see in the following sections.

#### 3.3.1 Defining via stochastic process (IBP)

Just as in the CRP, we can define our prior over infinite binary matrices using a simple stochastic process. In the Indian Buffet Process (IBP),  $N$  customers enter an Indian restaurant, one after another. Each customer is presented with a buffet of infinitely many dishes in a line. The first customer starts with the leftmost dish and samples  $\text{Poisson}(\alpha)$  dishes. The next customer samples each of the dishes sampled by the first customer with probability  $\frac{m_k}{i}$ , where  $m_k$  is the number of previous customers who have sampled dish  $k$  and  $i$  is the total number of previous customers. She then chooses an additional  $\text{Poisson}(\frac{\alpha}{i})$  new dishes to the right of the dishes sampled so far. This process repeats for all  $N$  customers.

We can use an  $N \times K$  binary matrix to represent the customers' dish choices (this matrix corresponds to the  $\mathbf{Z}$  matrix necessary for the latent feature model). This matrix will have a finite number of rows (customers) and a potentially infinite number of columns (dishes). An example of the matrix produced using  $\alpha = 10$  is shown in Figure 5. In this matrix, the first customer tried 17 dishes. The second customer sampled 7 of those dishes and three new dishes. The process was repeated for a total of 20 customers.

We can calculate the probability of a particular matrix being produced by the IBP as

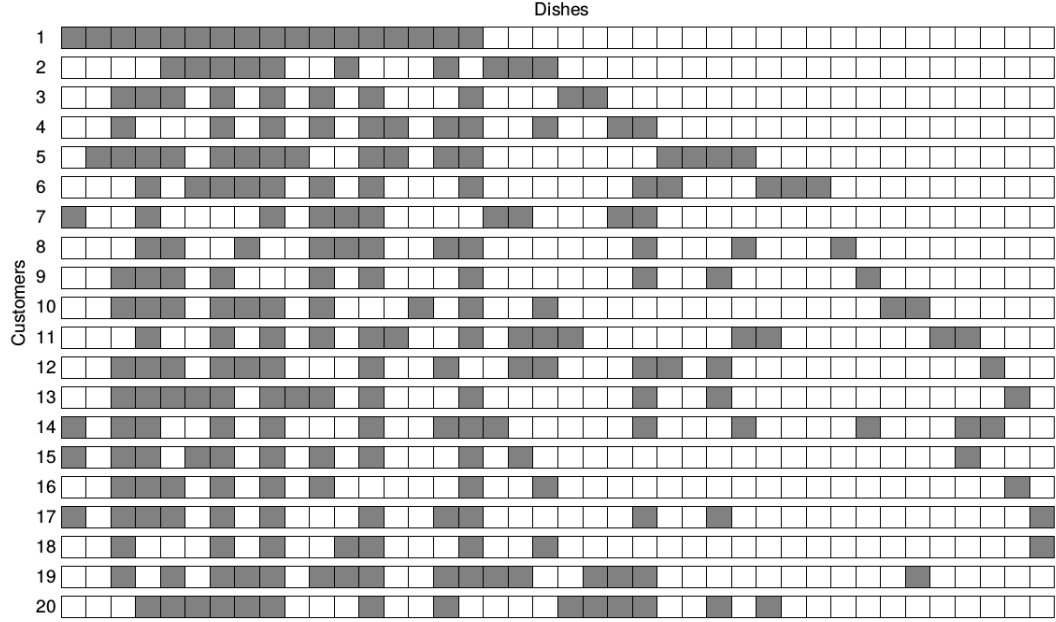


Figure 5: An example of the matrix  $\mathbf{Z}$  produced by running the IBP with  $\alpha = 10$ . The first customer tried 17 dishes. The second customer sampled 7 of those dishes and three new dishes. The process was repeated for a total of 20 customers.

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

where  $K+$  is the largest number dish that some customer previously sampled,  $K_1^{(i)}$  indicates the number of new dishes sampled by the  $i$ th customer and  $H_N$  is the  $N$ th harmonic number.

### 3.3.2 Defining as limit of finite model

Alternatively, we can define a prior over infinite binary matrices as the limit of a prior over finite binary matrices. In this section, we assume a model with a finite number of features  $K$ , and then take the limit as  $K \rightarrow \infty$ . We have  $N$  objects, each with  $K$  features. The indicator variable  $z_{ik}$  indicates whether object  $i$  possesses feature  $k$ . Each object independently possess feature  $k$  with probability  $\pi_k$ . Under this model, the probability of a matrix  $\mathbf{Z}$  given  $\pi = \{\pi_1, \pi_2, \dots, \pi_K\}$  is

$$P(\mathbf{Z}|\pi) = \prod_{k=1}^K \prod_{i=1}^N P(z_{ik}|\pi_k) = \prod_{k=1}^K \pi_k^{m_k} (1 - \pi_k)^{N - m_k} \quad (2)$$

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

In order to do Bayesian inference, given this likelihood, we need to define a prior on  $\pi$ . In the paper, they let each  $\pi_k$  be drawn from a Beta distribution with parameters  $r, s$ . Letting  $r = \frac{\alpha}{K}$  and  $s = 1$ , we have the

following probability model:

$$\begin{aligned}\pi_k | \alpha &\sim \text{Beta}\left(\frac{\alpha}{K}, 1\right) \\ z_{ik} | \pi_k &\sim \text{Bernoulli}(\pi_k)\end{aligned}\tag{3}$$

Given this model, we can integrate out  $\pi$  and find the marginal probability for  $\mathbf{Z}$ :

$$\begin{aligned}P(\mathbf{Z}) &= \prod_{k=1}^K \int \prod_{i=1}^N P(z_{ik} | \pi_k) P(\pi_k) d\pi_k \\ &= \prod_{k=1}^K \frac{\frac{\alpha}{K} \Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}\end{aligned}\tag{4}$$

We will now explore what happens to these equations as we take the limit  $K \rightarrow \infty$ .

#### Note: Equivalence classes

Thus far, we have talked about the probability of a specific feature matrix  $\mathbf{Z}$ . What we really want to define is the probability of a specific, exchangeable equivalence class of feature matrices. In the CRP model, we said that the order in which customers were assigned to tables did not matter. Every ordering that led to the same partition of the customers into tables was equivalent. These orderings formed an equivalence class, and we defined probabilities of partitions of customers, not specific orderings of assigning customers to tables.

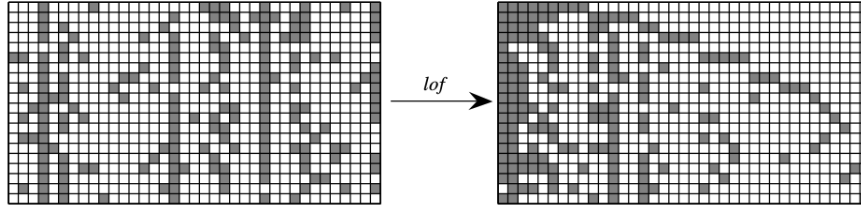


Figure 6: The transformation of an arbitrary matrix to its *left-ordered form* via the many-to-one *lof()* function.

We want to define a similar equivalence classes for binary matrices, so that we can talk about the probability of equivalence classes instead of individual matrices. The authors introduced an equivalence class based on a binary matrix's *left-ordered form*. After applying the many-to-one *lof()* function, all matrices in the same equivalence class will collapse to an identical matrix. The *lof()* function is shown in Figure 6.

Using Eq. 4, we can define the probability of an equivalence class of matrices  $[\mathbf{Z}]$  as

$$\begin{aligned}P([\mathbf{Z}]) &= \sum_{\mathbf{Z} \in [\mathbf{Z}]} P(\mathbf{Z}) \\ &= \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^K \frac{\frac{\alpha}{K} \Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}\end{aligned}\tag{5}$$

Taking the limit of Eq. 5 as  $K \rightarrow \infty$ , we are left with the following expression for  $P([\mathbf{Z}])$ :

$$P([\mathbf{Z}]) = \frac{\alpha^{K+}}{\prod_{h=1}^{2^{N-1}} K_h!} \exp\{-\alpha H_N\} \prod_{k=1}^{K+} \frac{(N - m_k)!(m_k - 1)!}{N!} \quad (6)$$

where  $K+$  is the number of columns for which there is at least one data point with that feature and  $H_N$  is the  $N$ th harmonic number. Further details of these derivations can be found in Griffiths et al. (2011).

Note that this form is slightly different than the probability introduced by the IBP in Eq. 1. However, the probability in Eq. 1 was defined for a specific matrix, not over  $\text{lof}()$  equivalence classes. The matrices produced by the IBP are not in left-order form, but are also not random, because customers choose dishes starting at the left side of the buffet. Therefore, there are  $\frac{\prod_{i=1}^N K^{(i)}!}{\prod_{h=1}^{2^{N-1}} K_h!}$  matrices generated by the IBP that map to the same left-ordered form. When this number is used to find the corresponding  $P([\mathbf{Z}])$  for Eq. 1 from the IBP, the result will be identical to Eq. 6.

### 3.3.3 Properties of IBP

We have now defined two processes to construct a prior over our infinite binary feature matrix  $\mathbf{Z}$ . Defined in this way,  $\mathbf{Z}$  has some interesting properties:

- The effective dimension of the model,  $K+$  grows as  $\text{Poisson}(\alpha H_N)$ .
- The number of features possessed by each object follows a  $\text{Poisson}(\alpha)$  distribution.
- The matrix  $\mathbf{Z}$  will remain sparse as  $K \rightarrow \infty$ . The expected number of entries in  $\mathbf{Z}$  is  $N\alpha$ .

## 3.4 Inference in the IBP Model

From the above sections, we know how to construct a prior over infinite, sparse, binary matrices that represent the presence/absence of an unbounded number of features in our  $N$  data points. The next important question to ask is whether inference is tractable under such an infinite model.

Given observed data  $\mathbf{X}$ , the authors introduce a Gibbs sampler for inferring the posterior  $P(\mathbf{Z}|\mathbf{X})$  from our derived prior  $P(\mathbf{Z})$  and a likelihood model. To sample from the desired posterior using Gibbs sampling, we need to compute the conditional distributions  $P(z_{ik} = 1|\mathbf{Z}_{-(ik)})$ . Because the IBP is exchangeable, we choose the current customer  $i$  to be the last customer to visit the buffet. This gives us

$$P(z_{ik} = 1|\mathbf{z}_{-i,k}) = \frac{m_{-i,k}}{N} \quad (7)$$

for any  $k$  such that  $m_{-i,k} > 0$ . This leads to the following inference procedure:

1. Initialize with an arbitrary binary matrix
2. For each row (customer) in the matrix,  $i$ 
  - (a) For each column (dish) in the matrix,  $k$ 
    - i. If  $m_{-i,k} > 0$ , set  $z_{ik}=1$  with probability given by Eq. 7 and incorporate the likelihood  $P(\mathbf{X}|\mathbf{Z})$ .
    - ii. Otherwise, delete column  $k$
  - (b) At the end of each row, add  $M$  new columns of 1's in that row, where  $M \sim \text{Poisson}(\frac{\alpha}{N})P(\mathbf{X}|\mathbf{Z})$ .

After a sufficient burn-in period, the matrix generated will be sampled from the posterior  $P(\mathbf{Z}|\mathbf{X})$ .

### 3.5 Examples: Linear-Gaussian Latent Feature Model

The authors apply the IBP prior to a linear-Gaussian latent feature model with binary features. To derive the likelihood term, they start with a finite model, and take the model's infinite limit. In the finite model, each data point  $\mathbf{x}_i$  is represented by a  $D$ -dimensional observable properties vector. Given a  $K$ -dimensional binary vector  $\mathbf{z}_i$ , these properties are generated from a Gaussian with mean  $\mathbf{z}_i \mathbf{A}$  and diagonal covariance matrix  $\Sigma_x = \sigma_x^2 \mathbf{I}$ , where  $\mathbf{A}$  is a  $K \times D$  matrix of weights.

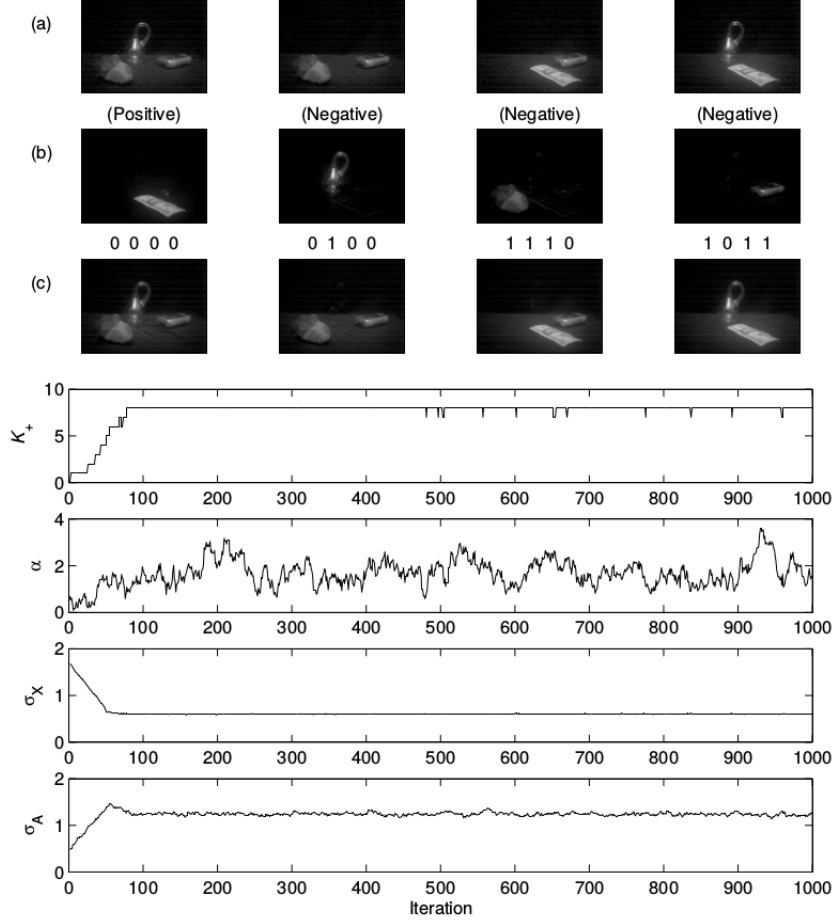


Figure 7: Data and results for the application of infinite linear-Gaussian model to images. (a) Four sample images from the data set. (b) The posterior mean of the weights for the four most frequent binary features after 1000 samples. These features correspond almost perfectly to the presence/absence of the four objects. (c) Reconstructions of the images in (a) using the binary features inferred for the images. The lower panels show the values of the parameters/hyperparameters throughout iterations of Gibbs sampling.

Our likelihood function  $P(\mathbf{X}|\mathbf{Z})$  is therefore Gaussian:

$$P(\mathbf{X}|\mathbf{Z}, \mathbf{A}, \sigma_x) = \frac{1}{(2\pi\sigma_x^2)^{(ND/2)}} \exp\left\{-\frac{1}{2\sigma_x^2} \text{tr}((\mathbf{X} - \mathbf{Z}\mathbf{A})^T(\mathbf{X} - \mathbf{Z}\mathbf{A}))\right\} \quad (8)$$

where  $\text{tr}()$  is the trace of a matrix. They then proceed to integrate out the model parameter  $\mathbf{A}$  by defining a Gaussian prior  $P(\mathbf{A})$  with hyperparameter  $\sigma_a$ . This leaves us with a likelihood function which is the



marginal likelihood for a Bayesian linear regression model. The authors take the limit of this likelihood function as the number of columns in  $\mathbf{Z} \rightarrow \infty$ . and derive a form of the infinite likelihood that only depends on  $K+$ , the number of columns with instantiated data, which will always be finite given finite data. The full derivation can be seen in Griffiths et al. (2011).

Given an infinite likelihood model  $P(\mathbf{X}|\mathbf{Z})$ , we can apply our infinite IBP prior  $P(\mathbf{Z})$  to find the model posterior using Bayes rule and posterior inference. Using the Gibbs sampler defined in Section 3.4, we can sample from the posterior  $P(z_{ik}|\mathbf{X}, \mathbf{Z}_{-(i,k)}, \sigma_x, \sigma_a) \propto p(\mathbf{X}|\mathbf{Z}, \sigma_x, \sigma_a)P(z_{ik}|\mathbf{Z}_{-(i,k)})$ . The authors approximate the distribution over the number of new features using truncation, computing  $K_1^{(i)}$  up to reasonable bound.

The authors then apply this linear-Gaussian model to recover latent structure in image data. The dataset consists of one hundred  $240 \times 320$  pixel images of four objects: a \$20 bill, a Klein bottle, a prehistoric handaxe, and a cellular phone. The objects are placed at a fixed location in each image, with some variation caused by hand placement. The data vector  $\mathbf{x}_i$  for the  $i$ th image was a 100-dimensional vector consisting of the weights of the mean image and the first 99 principle components of the  $i$ th image. In this setup, each object is a latent feature  $z_{ik}$  responsible for generating the observed pixel values in the image.

The results of posterior inference are shown in Figure 7 for the first 1000 iterations of MCMC. Besides using the Gibbs sampler to infer the matrix  $\mathbf{Z}$ , the authors add Metropolis steps to estimate  $\alpha$ ,  $\sigma_x$ , and  $\sigma_a$ . The values of these hyperparameters/parameters are shown in Figure 7 (bottom). Despite knowing that there are four features responsible for generating the pixels, the model converges on seven features. This is not surprising; nonparametric models will tend to overestimate the number of parameters in a scenario where the true number of parameters is fixed.

Figure 7 (a) shows example images from the dataset. Figure 7 (b) shows the posterior mean of  $\mathbf{a}_k$  for the four most frequent features in the 1000 iteration. These features appear to perfectly map to the presence/absence of the four objects when appropriately negated. The authors claim that the other three features discovered code for slight variations in location amount the objects. Figure 7 (c) shows the feature vector  $\mathbf{z}_i$  for four test images along with the posterior means of the reconstructions of the images in (a) using the discovered features.

The authors conclude their paper with a comprehensive list of applications of the IBP to solve a diverse set of problems from social science, genetics, networks, infinite Markov Models, and causality inference. The references for these examples are included in Griffiths et al. (2011).

## References

- Blei, D. M., Griffiths, T. L., Jordan, M. I., Blei, D. M., and Jordan, M. I. (2010). The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies. *J. ACM Journal of the ACM*, 57(7).
- Gershman, S. J. and Blei, D. M. (2011). A Tutorial on Bayesian Nonparametric Models.
- Griffiths, T. L., Edu, T. G., Ghahramani, Z., and Uk, Z. C. A. (2011). The Indian Buffet Process: An Introduction and Review. *Journal of Machine Learning Research*, 12:1185–1224.
- Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2009). Distributed Algorithms for Topic Models. *Journal of Machine Learning Research*, 10.
- Raykov, Y. P., Boukouvalas, A., Baig, F., Little, M. A., Ermani, M., and Battistin, L. (2016). What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. *PLOS ONE*, 11(9):e0162259.