Extending The Indian Buffet Process to Incorporate Pairwise Distance

Arthur Lui

A selected Project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David B. Dahl, Chair
Gilbert W. Fellingham
David A. Engler

Department of Statistics

Brigham Young University

April 2015

ABSTRACT

Extending The Indian Buffet Process to Incorporate Pairwise Distance

Arthur Lui
Department of Statistics, BYU
Master of Science

One challenge in latent feature modelling is that the number of latent features has to be predetermined by the researcher or subject experts. Bayesian nonparametrics offers methods to learn the number of latent features while discovering the latent features themselves. The Indian buffet process (IBP) provides a distribution over sparse binary matrices of infinite dimensions, and can be used as a prior for feature matrices in latent feature models. The IBP assumes exchangeability, but this is not always an appropriate assumption as data may often be inter-related. We, therefore propose a distribution, based on the Indian buffet process, which will incorporate pairwise similarity information between observations. Gershman et al. (2012) have proposed the distance dependent Indian buffet process (dd-IBP), which indeed incorporates distance information into the IBP. Our proposed distribution will incorporate distance information in a different manner, using attraction information comparable to that used in the Ewens-Pitman attraction (EPA) distribution developed by Dahl et al. (2014). Our goal is to propose a distribution which reduces to the IBP in base cases, but uses attraction information to provide a distribution which has similar properties as the EPA.

Keywords: Bayesian nonparametrics, Indian buffet process, latent feature models, Chinese restaurant process, distance-dependent Chinese restaurant process, Ewens-Pitman attraction distribution

ACKNOWLEDGMENTS

I would like to first of all thank Dr. David B. Dahl, my research adviser. The nature of research in statistics is that researchers want to find answers to unsolved problems. So, I am grateful for a mentor that encourages me to search, think, and study on my own when I cannot find an answer elsewhere. I also express gratitude to the department faculty who have taught me at Brigham Young University. Their enthusiasm for the subject of statistics and my success have often inspired me to press on with hope when I am discouraged by my research and performance in class. They have taught me a great deal and given me an opportunity to see meaningful applications of statistics. This instills, in me, aspirations to contribute to my community through the subject. Finally, I express gratitude to my father and mother, who have always been supportive of me and all my fields of studies, including music. They have never tried to persuade me to study something I did not enjoy. Because of their unconditional love and support, I believe I have been able to enjoy a fulfilling and balanced life thus far.

CONTENTS

CHAPTER 1

INTRODUCTION

The goal of this project is to define a distribution over sparse binary matrices of infinite dimensions. Specifically, we want to extend the Indian buffet process (IBP) to include distance information between observations.

The IBP provides a distribution for binary matrices of infinite dimensions. They are a good choice of prior distribution for feature matrices in infinite latent feature models as their dimensions can be learned from the data and do not have to be predetermined.

Implementations of the IBP to include distance information have been studied and introduced by authors such as Gershman et al. (2012). The distance-dependent Indian buffet process (ddIBP) is one such implementation. It reduces to the regular IBP under certain conditions, and preserves many of the properties of the IBP. We will propose a new distribution which makes use of attraction information that Dahl et al. (2014) use in the Ewen-Pitman attraction (EPA) distribution. We will draw comparisons between the ddIBP and the proposed distribution. We hope to preserve as many features of the IBP as possible. The proposed distribution will also have a p.m.f which does not require the enumeration of all possible connectivity matrices and ownership vectors, as in the ddIBP.

CHAPTER 2

LITERATURE REVIEW

In this section, we lay the groundwork for our proposed distribution by reviewing important processes and distributions. We will discuss the Chinese restaurant process, the distance dependent Chinese restaurant process (Blei and Frazier 2011), the Ewens-Pitman attraction distribution (Dahl et al. 2014), the Indian buffet process (Griffiths and Ghahramani 2011), and the distance dependent Indian buffet process (Gershman et al. 2012).

2.1 The Chinese Restaurant Process

Bayesian nonparametrics provides flexible models which can determine underlying structure from data. For instance, the Chinese restaurant process (CRP) can serve as a prior on partition distrbutions in a mixture model. The size and number of partitions can be learned from the data in such a model.

The Chinese restaurant process describes the Dirichlet process, which creates partition distributions. The partition distribution generated by the CRP is the Ewens-Pitman distribution. The CRP can be described as follows. A number of customers (observations), n, enter a Chinese restaurant to be seated one at a time. Let z_i be the table (cluster) number to which each customer gets assigned, such that $z_i \in \{1, .., n\}$, for $i = 1, ..., n$. Let $z_{1:(i-1)}$ denote the vector of table numbers to which customers $1, ..., i-1$ are assigned. Then,

$$P(z_i = k | z_{1:(i-1)}, \alpha) \propto \begin{cases} n_k, & \text{if} \quad k \leq K \\ \alpha, & \text{if} \quad k = K+1 \end{cases} \tag{2.1}$$

where the normalizing constant is $(\alpha + i - 1)^{-1}$, n_k is the number of customers in table k before seating customer i, K is the number of occupied tables before seating customer i, and α is a mass parameter which determines the number of tables that will eventually be occupied by the n customers. The larger α is, the greater the final number of occupied tables will be. Let π_n represent the partition $\{S_1, ..., S_{q_n}\}$, where $q_n \in \{1, ..., n\}$ is the number of partitions and each S_i is a set representing the cluster of customers seated at table i, π_n will have the properties: (1) $S_i \cap S_j = \emptyset$ for $i \neq j$ (sets are mutually exclusive), (2) $\underset{S\in\pi_n}{\cap} S = \{1, ..., n\}$ (sets are exhaustive), and (3) $S_i \neq \emptyset$, $\forall i \in \{1, ..., q_n\}$ (no sets are empty). The probability mass function for the Ewens-Pitman distribution is

$$P(\pi_n) = \prod_{S\in\pi_n} \frac{\alpha\Gamma(|S|)}{\alpha^{(n)}}, \tag{2.2}$$

where $\alpha^{(n)} = \prod_{i=1}^{n}(\alpha + i - 1)$, $\Gamma(.)$ is the Gamma function, and $|S|$ denotes the cardinality of the set S.

Gibbs Sampler for CRP

Gibbs samplers to generate this process have been studied extensively. A valuable list and comprison of algorithms to draw from the posterior with a CRP prior was compiled by Neal (2000). An implementation of some of these algorithms in **R** and **Scala** can be found at my github account: https://github.com/luiarthur/auxGibbs/tree/master/scala.

2.2 Distance Dependent Chinese Restaurant Process

A deficiency of the CRP is that large clusters get larger, while small clusters stay small. Often, clustering observations according to a distance metric measured on the observations in more appropriate. This has motivated the development of algorithms that make use of the ideas in the CRP and incorporate distance information.

Researchers such as Blei & Frazier have extended CRP to include distance information. The distribution they have proposed is the distance dependent Chinese restaurant process (ddCRP). Given a distance matrix D and a mass parameter α, the probability of customer i being "assigned" to sit with customer j is

$$P(c_i = j|D, \alpha) \propto \begin{cases} f(d_{ij}), & \text{if} \quad j \neq i \\ \alpha, & \text{if} \quad i = j \end{cases} \tag{2.3}$$

where $f(d_{ij})$ is a decay function with the properties: (1) $f(\cdot) \geq 0$, (2) $f(\infty) = 0$, and (3) $f(\cdot)$ is non-increasing. The normalizing constant is again $(\alpha + i - 1)^{-1}$. Customers assigned to sit together at a table form a cluster. The resulting distribution is a partition distribution that includes pairwise distance information. The p.m.f. for the resulting partition is calculated by enumerating all assignments that map to a particular partition. So, algorithms like Metropolis-Hastings where the calculation of the p.m.f. is required cannot be implemented when using the ddCRP. However, Blei and Frazier (2011) describe a Gibbs sampler to sample from the ddCRP, which can be used to sample from posterior distributions when the prior distribution is the ddCRP.

2.3 Ewens-Pittman Attraction Distribution

While the ddrcp is intuitive and the construction of the distribution is relatively straight forward, it does not preserve certain properties of the CRP. For instance, the number of partitions and size of partitions are altered by distance information. That is, if the regular CRP would produce 5 partitions on average, the ddcrp may not produce the same number of partitions on average. Moreover, the p.m.f. for the ddcrp cannot be explicitly written out, but only implied through an algorithm. Consequently, MCMC algorithms like

Metrolois-Hastings cannot be used in obtaining posterior distributions where the prior is a ddcrp. These issues motivate the need for a distribution that preserves more properties of the original CRP and whose p.m.f. can be explicitly written.

The Ewens-Pitmann Attraction (EPA) distribution is a partition distribution that incorporates pairwise distance information (Dahl et al. 2014). Before introducing the probability mass function, we will review some notation. Let π_n be a discrete partition distribution, as defined in the previous section. Let the permutation $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)$ be the order in which each item is allocated, where the t^{th} item to be allocated is σ_t. This is not necessarily the order of the n items in the dataset. In addition, at time $t > 1$, let $\pi(\sigma_1, ..., \sigma_{t-1})$ represent the current partition created from allocating $\sigma_1, ..., \sigma_{t-1}$. Note that the complete partition $\pi_n = \pi(\sigma_1, ..., \sigma_n)$.

The EPA distribution incorporates pairwise distance information. A possible metric for measuring distance is the Euclidean metric. Let d_{ij} denote the distance between two items, i and j. Let the function $\lambda(i, j)$ represent the similarity of items i and j (i.e. how "close" two items are, where a larger value indicates that the two items are closer together). A large class of similarity functions can be represented as a function of the distance information. That is $\lambda(i, j)$ can be written as $f(d_{ij})$ in many cases, where $f(\cdot)$ is a non-increasing function. For instance, $\lambda(i, j)$ can be ${d_{ij}}^{-\tau}$, where $\tau > 0$ and is called the *temperature* and can dampen or accentuate the effect of distance.

The EPA distribution is also defined by a mass parameter, $\alpha > 0$, and a discount parameter, $\delta \in [0, 1)$. The probability mass function for a partition distribution π_n following the EPA distribution can be defined as follows:

$$p(\pi_n|\alpha, \delta, \lambda, \boldsymbol{\sigma}) = \prod_{i=1}^{n} p_t(\alpha, \delta, \lambda, \pi(\sigma_1, ..., \sigma_{t-1})) \tag{2.4}$$

where $p_t(\alpha, \delta, \lambda, \pi(\sigma_1, ..., \sigma_{t-1}))$ is defined as:

$$P(\sigma \in S|\alpha, \delta, \lambda, \pi(\sigma_1, ..., \sigma_{t-1})) = \begin{cases} \dfrac{t-1-\delta q_{t-1}}{\alpha+t-1} \cdot \dfrac{\sum_{\sigma_s \in S} \lambda(\sigma_t, \sigma_s)}{\sum_{s=1}^{t-1} \lambda(\sigma_t, \sigma_s)} & \text{for } S \in \pi(\sigma_1, ..., \sigma_{t-1}) \\ \dfrac{\alpha+\delta q_{t-1}}{\alpha+t-1} & \text{for } S = \emptyset \end{cases} \tag{2.5}$$

Note that the ratio of sums in (2.5) represents the proportion of *total attraction* of item σ_t to the items allocated to subset S.

2.4 The Indian Buffet Process

The purpose of the review so far is to show the methods that have been used to incorporate distance information into the CRP. The goal of this project is to use similar ideas of incorporating distance information into the Indian buffet process (IBP), which is a prior distribution for matrices in another typical Bayesian nonparametrics model - the latent feature model.

One key problem in recovering the latent structure responsible for generating observed data is determining the number of latent features. The Indian Buffet process (IBP) provides a flexible distribution for sparse binary matrices with infinite dimensions (i.e. finite number of rows, and infinite number of columns). When used as a prior distribution in a latent feature model, the IBP can learn the number of latent features generating the observations because it can draw binary matrices which have a potentially infinite number of columns. We will use the IBP as a prior distribution in a Gaussian latent feature model to recover the latent structures generating the observations (Griffiths and Ghahramani 2011).

The IBP is a distribution for sparse binary matrices with a finite number of rows and potentially an infinite number of columns. The process of generating a realization from the IBP can be described by an analogy involving Indian buffet restaurants.

Let Z be an $N \times \infty$ binary matrix. Each row in Z represents a customer who enters an Indian buffet and each column represents a dish in the buffet. Customers enter the restaurant one after another. The first customer samples an $r =$Poisson(α) number of dishes, where $\alpha > 0$ is a mass parameter which influences the final number of sampled dishes. This is indicated in by setting the first r columns of the first row in Z to be 1. The other values in the row are set to 0. Each subsequent customer samples each previously sampled dish with probability proportional to its popularity. That is, the next customer samples dish k with probability m_k/i, where m_k is the number of customers that sampled dish k, and i is the current customer number (or row number in Z). Each customer also samples an additional Poisson(α/i) number of new dishes. Once all the N customers have gone through this process, the resulting Z matrix will be a draw from the Indian buffet process with mass parameter α. In other words, $Z \sim$ IBP(α). Note that $\alpha \propto K_+$, where K_+ is the final number of sampled dishes (occupied columns). Figure 2.1 shows a draw from an IBP(10) with 50 rows. The white squares are 1, indicating that a dish was taken; black squares are 0, indicating that a dish was not taken.

The probability of any particular matrix produced from this process is

$$P(\boldsymbol{Z}) = \frac{\alpha^{K_+}}{\prod_{i=1}^{N} {K_1}^{(i)}!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N - m_k)!\,(m_k - 1)!}{N!}, \tag{2.6}$$

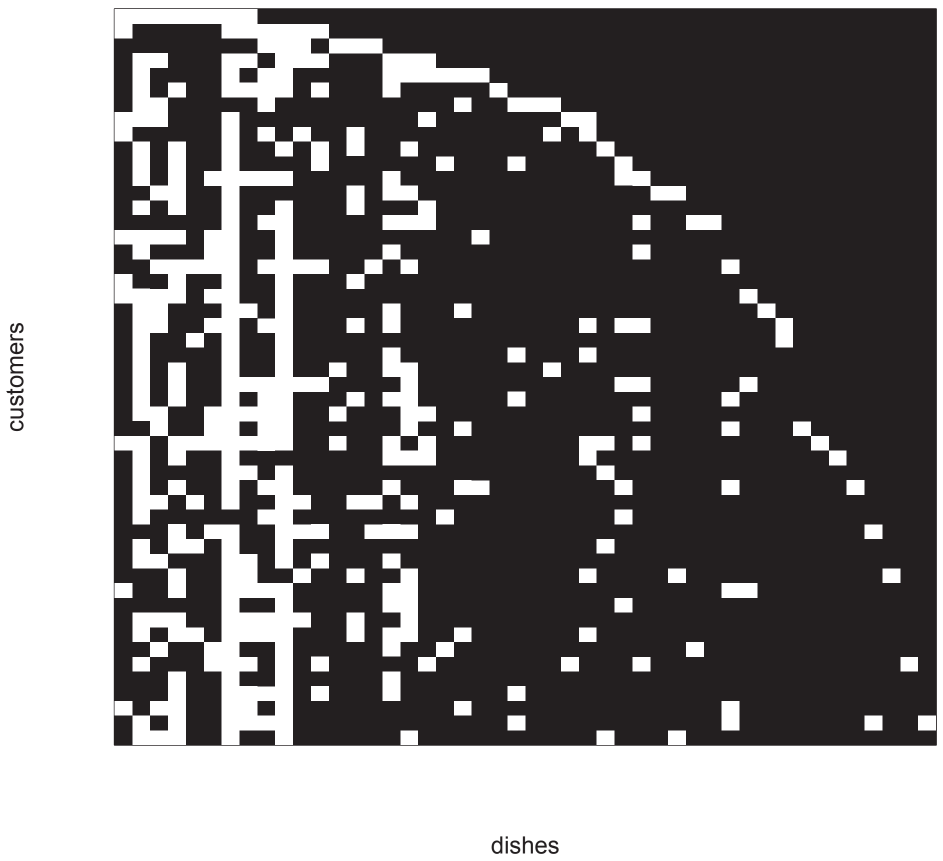


Figure 2.1: Random draw from the Indian buffet process with $\alpha = 10$ and 50 rows

where H_N is the harmonic number, $\sum_{i=1}^{N} \frac{1}{i}$, K_+ is the number of non-zero columns in $\boldsymbol{Z}$, m_k is the k^{th} column sum of $\boldsymbol{Z}$, and $K_1^{(i)}$ is the "number of new dishes" sampled by customer i.

Gibbs Sampler for Indian Buffet Process

One way to get a draw from the IBP(α) is to simulate the process according to the description above. Another way is to implement a Gibbs sampler (Griffiths and Ghahramani 2011). We can implement a Gibbs sampler to draw from the IBP as follows:

1. Start with an arbitrary binary matrix of N rows
2. For each row, i,
   a) For each column, k,
   b) if $m_{-i,k} = 0$, delete column k. Otherwise,

c) set z_{ik} to 0

d) set z_{ik} to 1 with probability $P(z_{ik} = 1|\boldsymbol{z_{-i,k}}) = \frac{m_{-i,k}}{i}$

e) at the end of row i, add Poisson($\dfrac{\alpha}{N}$) columns of 1's

3. iterate step 2 a large number of times

We can likewise incorporate this Gibbs sampler to sample from the posterior distribution (Griffiths and Ghahramani 2011) P($\boldsymbol{Z}|\boldsymbol{X}$) where $\boldsymbol{Z} \sim$ IBP(α) by sampling from the complete conditional

$$P(z_{ik} = 1|\boldsymbol{Z_{-(ik)}}, \boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{Z})P(z_{ik} = 1|\boldsymbol{Z_{-(ik)}}). \tag{2.7}$$

The parameter α is often unknown, so it should be modeled. Note that the conjugate prior for α is a Gamma distribution. Using a Gamma distribution is appropriate since α is positive.

$$\begin{aligned} \boldsymbol{Z}|\alpha &\sim \text{IBP}(\alpha) \\ \alpha &\sim \text{Gamma}(a, b), \text{where } b \text{ is the scale parameter} \end{aligned}$$

$$\begin{aligned} p(\alpha|\boldsymbol{Z}) &\propto p(\boldsymbol{Z}|\alpha)p(\alpha) \\ p(\alpha|\boldsymbol{Z}) &\propto \alpha^{K_+}e^{-\alpha H_N}\alpha^{a-1}e^{-\alpha/b} \\ p(\alpha|\boldsymbol{Z}) &\propto \alpha^{a+K_+-1}e^{-\alpha(1/b+H_N)} \end{aligned}$$

$$\alpha|\boldsymbol{Z} \sim \text{Gamma}(a + K_+, (1/b + H_N)^{-1}) \tag{2.8}$$

Example: Linear-Gaussian Latent Feature Model with Binary Features

Suppose, we observe an $N \times D$ matrix $\boldsymbol{X}$, and we believe

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A} + \boldsymbol{E},$$

where $\boldsymbol{Z}|\alpha \sim IBP(\alpha)$, $\boldsymbol{A} \sim MVN(\mathbf{0}, {\sigma_A}^2\mathbf{I})$, $\boldsymbol{E} \sim MVN(\mathbf{0}, {\sigma_X}^2\mathbf{I})$, and $\alpha \sim$ Gamma(a, b), It has been shown by Griffiths and Ghahramani (2011) that

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{Z}) = & \frac{1}{(2\pi)^{ND/2}\sigma_X^{(N-K)D}\sigma_A^{KD}|\boldsymbol{Z}^T\boldsymbol{Z} + (\frac{\sigma_X}{\sigma_A})^2\mathbf{I}|^{D/2}} \\ & \exp\{-\frac{1}{2\sigma_X^2}tr(\boldsymbol{X}^T(\mathbf{I} - \boldsymbol{Z}(\boldsymbol{Z}^T\boldsymbol{Z} + (\frac{\sigma_X}{\sigma_A})^2\mathbf{I})^{-1}\boldsymbol{Z}))\boldsymbol{X}\} \end{aligned} \tag{2.9}$$

Now, we can use equation (2) to implement a Gibbs sampler to draw from the posterior $\boldsymbol{Z}|\boldsymbol{X}, \alpha$.

Data & Results

Each of ten students created a 6×6 binary image. Gaussian noise (mean=0, variance=.25) was added to each cell of the binary image to generate 10 6×6 images. These images were turned into 1×36 row vectors. All these vectorized images were stacked together to form one large 100×36 matrix. (Note that the design of these images were not known beforehand. The images could be letters, numbers, various patterns, etc.) Figure 2.2 displays the data from the ten students.

A Gibbs sampler was implemented to retrieve posterior distributions for $\boldsymbol{Z}$, $\boldsymbol{A}$, and α. The mass parameter α was initially set to 1, and the posterior for α was obtained by equation (3) with prior distribution $Gamma(3, 2)$. The parameters were chosen such that α was centered at 6 and had a variance of 12, because it is probable that there are many latent features. Equation(2) was used to retrieve the posterior distribution for $\boldsymbol{Z}$ (a collection of binary matrices). After 5000 iterations, the trace plot for the number of columns in the binary matrices drawn were plotted (See Figure 2.3). Diagnostics for a cell by cell trace plot could also be plotted, but may be too difficult to analyze as the dimensions of the matrices are changing, and the matrices are large. The execution time was approximately 4 hours. The number of columns in $\boldsymbol{Z}$ appears to have converged to 9. This means that the number of latent features discovered appears to be 9. This is reasonable as the image $\boldsymbol{X}$ is comprised

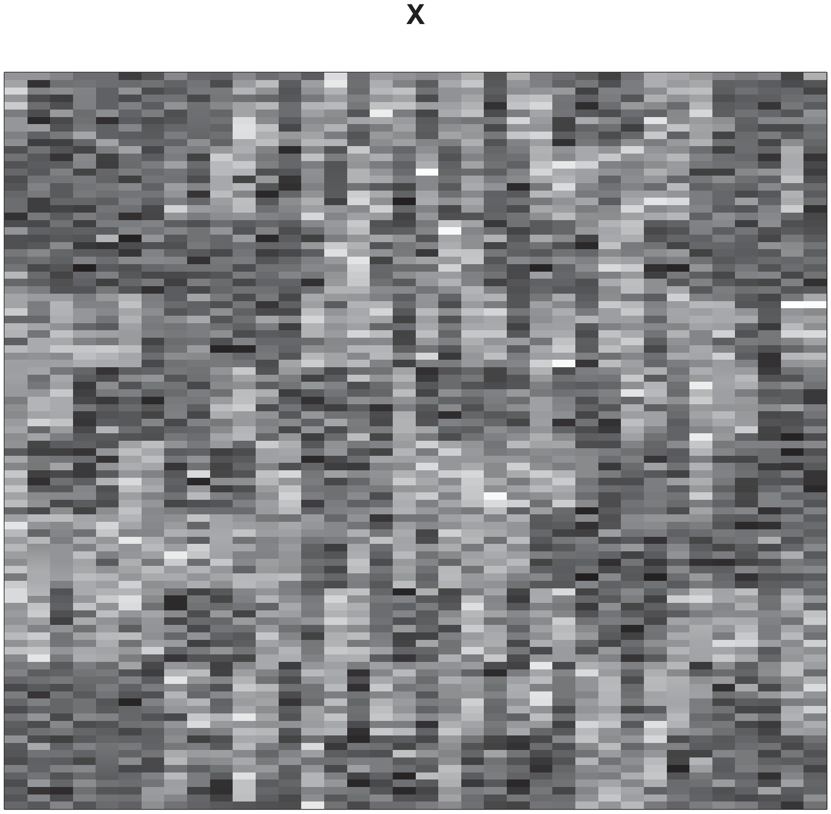


Figure 2.2: A 100×36 data matrix $\boldsymbol{X}$. Each row is a vectorized 6×6 binary image created by one of ten students with gaussian noise (mean=0, variance=.25) added to each cell.

of 10 students' images. A burn-in of the first 1000 draws were removed. Then, the 4000 $\boldsymbol{Z}$ matrices were superimposed, summed element by element, and divided by 4000. Cells that had values $> .9$ were set to 1; and 0 otherwise. This is not necessary, but this removes the columns that were not likely to exist. To elaborate, from the trace plot (Figure 2.3), we see that after burn-in, there is one instance where the number of columns in $\boldsymbol{Z}$ was 10. One instance out of 4000 is not significantly large enough to say that *that* latent feature is generating the observed data. So, the tenth column was removed. The resulting matrix, will be referred to as the posterior mean for $\boldsymbol{Z}$. The trace plot for α (Figure 2.4) shows that α appears to have converged, with mean = 2.08 and variance = .359. Figure 2.5 shows the posterior mean for $\boldsymbol{Z}$. We can interpret the matrix in the following way. All the observations are being generated by the first column (feature). The 11^{th} through 20^{th} observations in $\boldsymbol{X}$

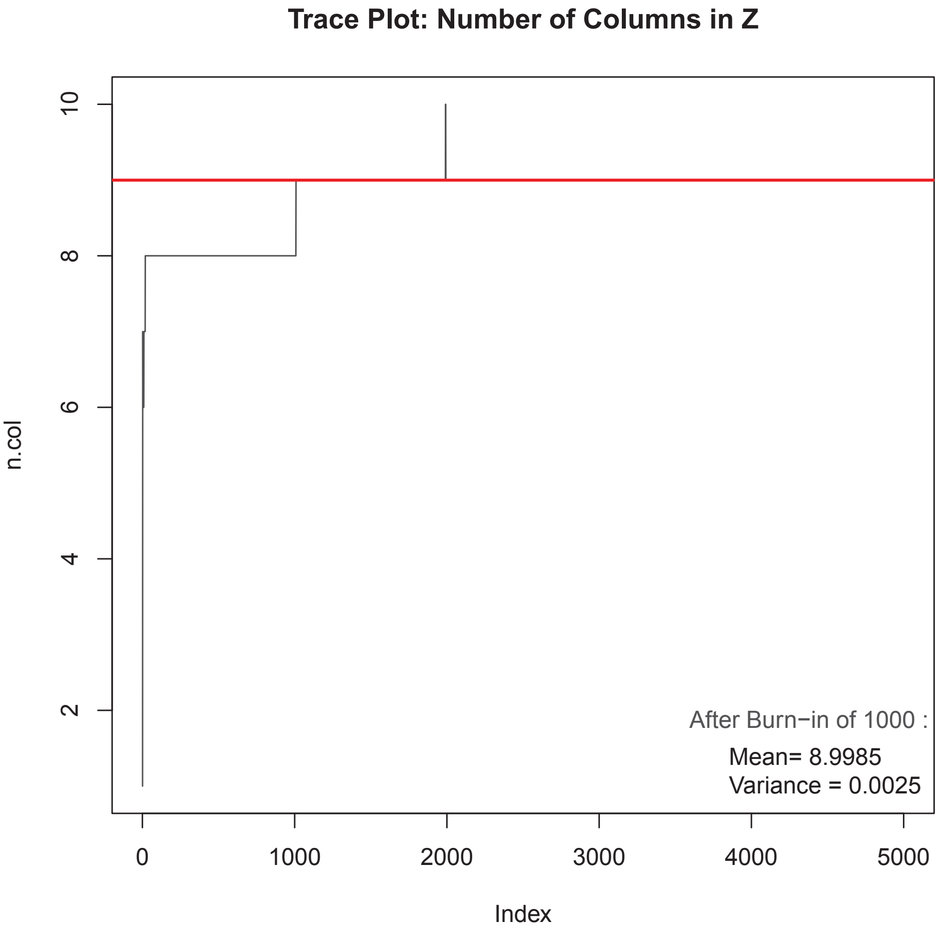


Figure 2.3: Trace plot of the number of columns in the matrices sampled from the posterior distribution $\boldsymbol{Z}|\boldsymbol{X}$

are being generated by the second column, etc. The posterior mean for $\boldsymbol{A}$ calculated as $E[\boldsymbol{A}|\boldsymbol{X},\boldsymbol{Z}] = (\boldsymbol{Z}^T\boldsymbol{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\boldsymbol{Z}^T\boldsymbol{X}$, and is shown in Figure 2.6 and Figure 2.7. Figure 2.6 shows the matrix in a 10×36 form; Figure 2.7 shows the matrix in 6×6 form. That is, each row in $\boldsymbol{A}$ was back-transformed into its matrix form.

Now, we can predict the latent features generating each observation by multiplying the posterior mean of $\boldsymbol{Z}$ by the posterior mean of $\boldsymbol{A}$. We will call this matrix the posterior mean of $\boldsymbol{ZA}$. Each row in this matrix will reveal the latent structures generating the observation

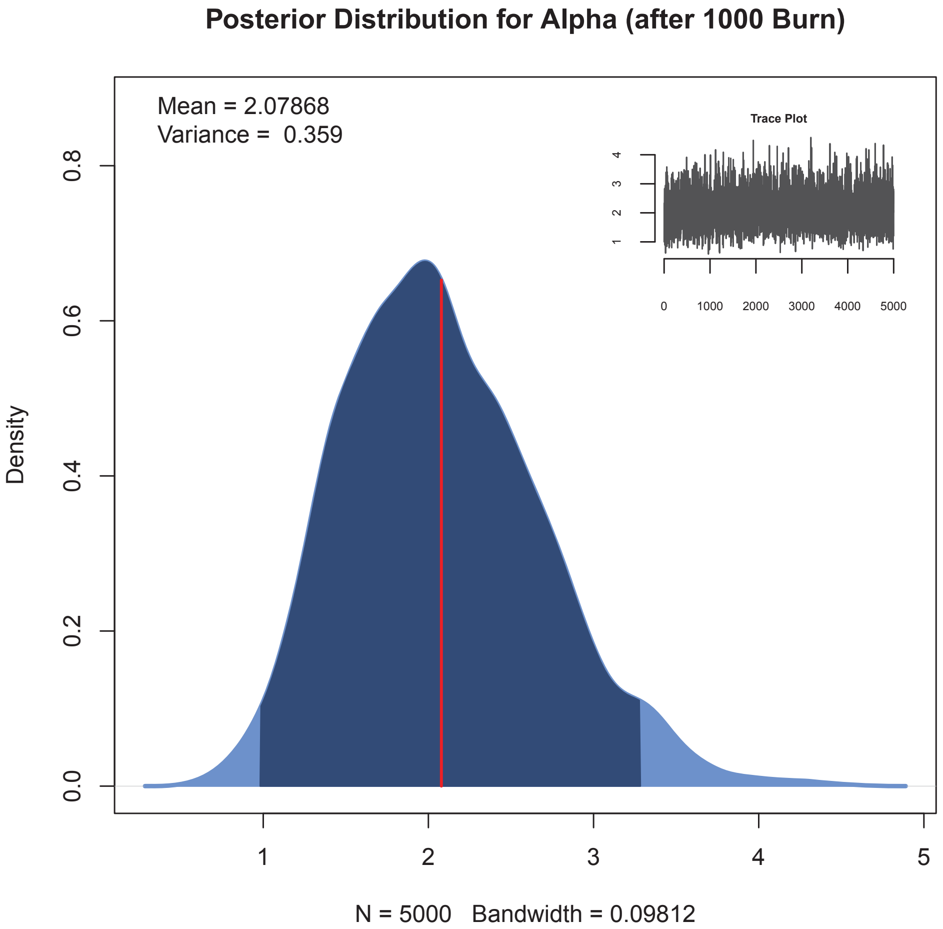


Figure 2.4: Posterior distribution for α with trace plot for α in the top right corner.

in the respective row of $\boldsymbol{X}$. Figure 2.8 shows the latent structure learned from the data. For each observation, the data and latent structures are layed side by side for a visual comparison. The features learned were similar to the images created by the ten students.

2.5 Distance Dependent Indian Buffet Process

Similarly to the CRP, the IBP has property of "making the rich richer". That is, features (dishes) get taken by observations with probability proportional to their popularity. It is sometimes more appropriate for observations to take on features based on their proximity to other observations. This has motivated the development of the IBP to include distance

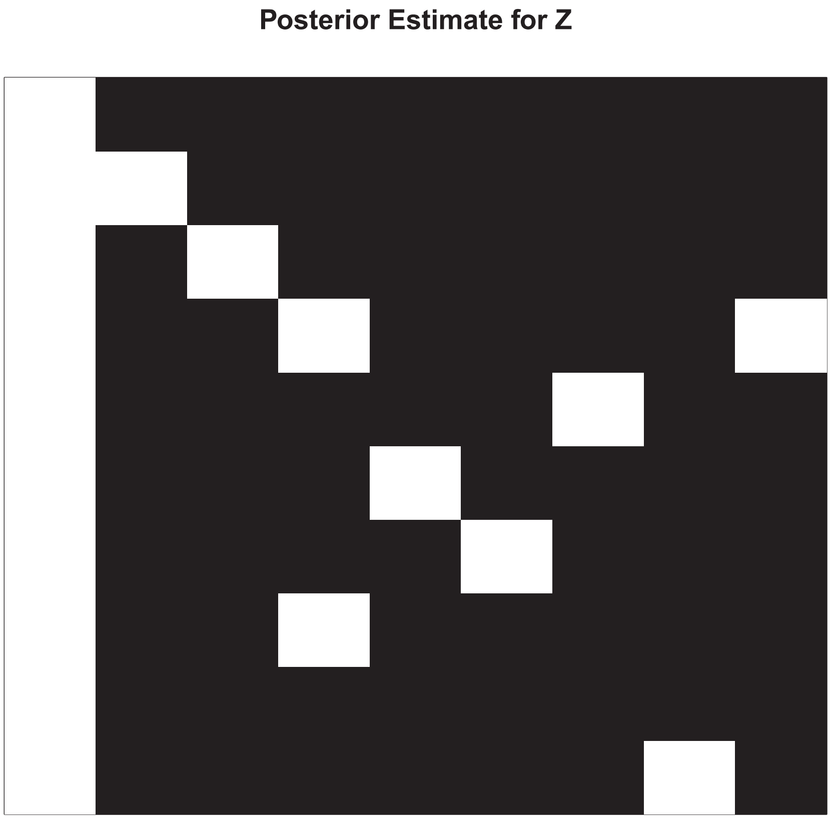


Figure 2.5: Posterior mean for $\boldsymbol{Z}$ computed by summing acoss all $\boldsymbol{Z}$ matrices drawn from the posterior distribution, and dividing each cell by the number of matrices drawn. Each cell was set to 1 if the value of the cell was greater than .8

information.

Currently, there exists a distance dependent Indian buffet process (ddIBP), constructed by Gershman et al. (2012). We wish to create a model similar to it, but that preserves fundamental properties of the IBP, and whose p.m.f. can be written out explicitly rather than implicitly as in the ddcrp. We will borrow ideas from Dahl et al. (2014) to develop this proposed distribution. We will first review the ddIBP.

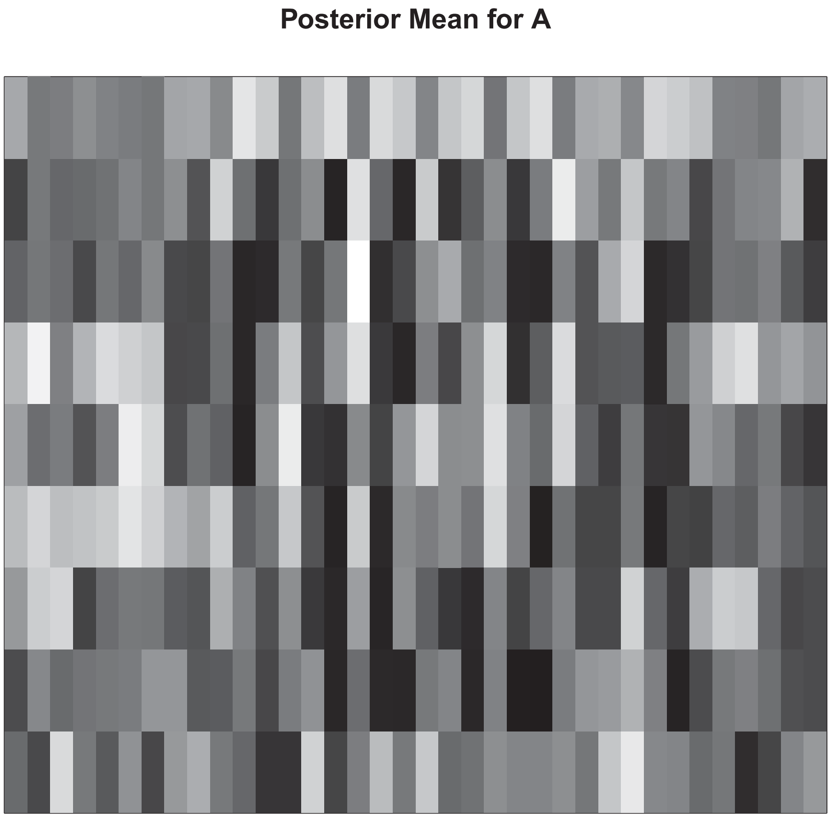


Figure 2.6: Posterior mean of A $=E[\boldsymbol{A}|\boldsymbol{X},\boldsymbol{Z}] = (\boldsymbol{Z}^T\boldsymbol{Z} + \frac{\sigma_X^2}{\sigma_A^2}\mathbf{I})^{-1}\boldsymbol{Z}^T\boldsymbol{X}$.

In the ddIBP, following the same analogy as in the IBP, two customers that are "closer" together in a given distance metric are more likely to share the same dishes (features). This is different from the IBP as new customers do not take new dishes according to their popularity. Terminology and notation for the ddIBP will here be introduced:

1. Dishes (columns of $\boldsymbol{Z}$) are identified by the natural numbers $\mathbb{N}$
2. $\mathcal{K}_i$: The set of dishes owned by customer i
   - $\mathcal{K}_i \subset \mathbb{N}$
   - $\mathcal{K}_i \cap \mathcal{K}_j = \emptyset$, for $i \neq j$, i.e. the sets, representing the dishes that different customers own, are disjoint

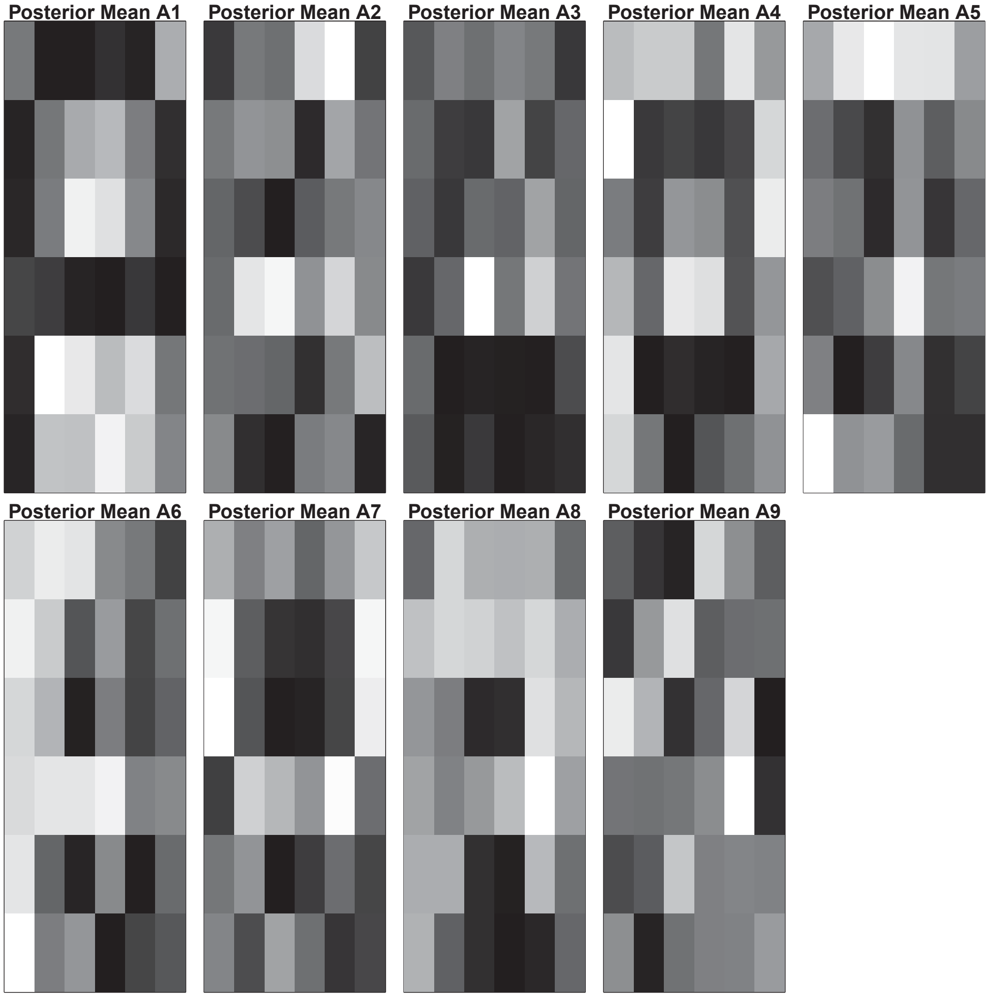


Figure 2.7: The latent features back-transformed to 6×6 images. These images are obtained by converting each row of the posterior mean of A into 6×6 images.

- $\lambda_i = |\mathcal{K}_i|$, the number of dishes owned by customer i
- $K = \sum_{i=1}^{N} \lambda_i$, the total number of owned dishes
- $K_{-i} = \cup_{j \neq i} K_j$, the set of all owned dishes excluding those of customer i

3. $\boldsymbol{C}$: The N × K connectivity matrix, which indicates how customers are "linked" to each other by dishes
    - $c_{ik} = j \implies$ customer i connects to customer j through dish k
4. $\boldsymbol{c^*}$: The ownership vector, $\boldsymbol{c^*_k} \in \{1, ..., N\}$ indicates the owner of dish k

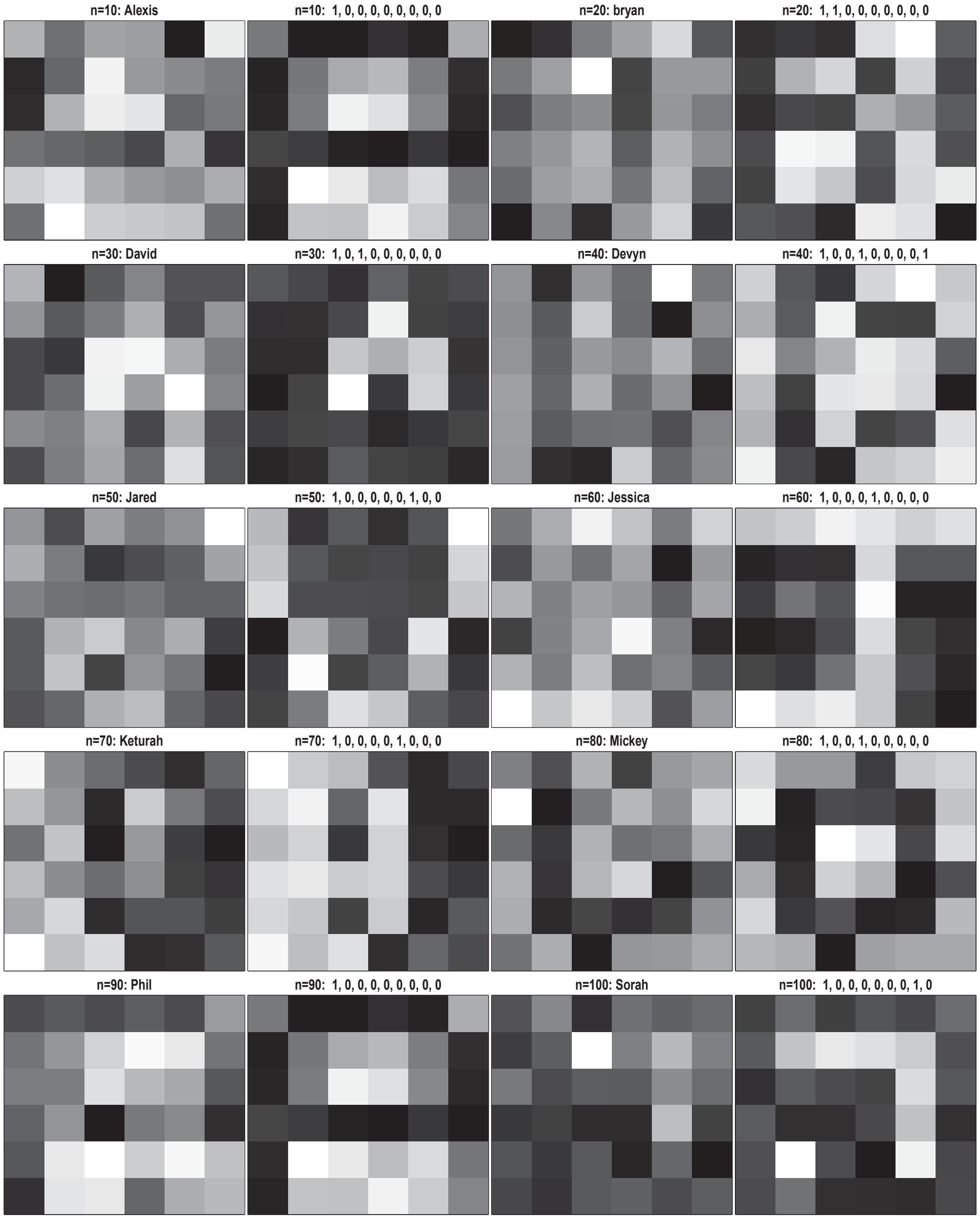


Figure 2.8: Estimated latent feature for each student obtained placed on the right of one observation from each student.

- $\boldsymbol{c^*_k} = i \implies k \in \mathcal{K}_i$

5. $\boldsymbol{D}$: The N × N distance matrix, where d_{ij} indicates the distance of customer j from customer i
6. $f : \mathbb{R} \rightarrow [0,1]$: The decay function, f maps the distance between customers to unity. The decay function has the properties:
   - $f(0) = 1$
   - $f(\infty) = 0$
   - $f(\cdot)$ is monotone decreasing
7. $\boldsymbol{A}$: The N × N normalized proximity matrix is defined as $a_{ij} = f(d_{ij})/h_i$, where $h_i = \sum\limits_{j=1}^{N} f(d_{ij})$

To generate an observation from the ddIBP with mass parameter α, decay function f, and distance matrix $\boldsymbol{D}$ for N observations, we use the following algorithm (Gershman et al. 2012):

1. Set $\lambda_0 := 0$
2. For $i = 1 : N$
   - draw $\lambda_i \sim \text{Poisson}(\alpha/h_i)$
   - set $\mathcal{K}_i := \{\sum\limits_{j=0}^{i-1} \lambda_j, ..., \sum\limits_{j=0}^{i-1} \lambda_j + \lambda_i\}$
   - for each $k \in \mathcal{K}_i$, set $\boldsymbol{c^*_k} := i$
3. For $i = 1 : N$
   - for $k = 1 : K$, assign customer i to connect to customer j by dish k with probability $\text{P}(c_{ik} = j | \boldsymbol{D}, f) = a_{ij}$, where $j = 1, ..., N$
4. Generate $\boldsymbol{Z}$ matrix:
   - for each customer i
     - for each dish k, if dish k is reachable by customer i through other customers, or if customer i owns the dish, then $z_{ik} := 0$

The feature matrix, $\boldsymbol{Z}$ is computed deterministically from the connectivity matrix $\boldsymbol{C}$ and ownership vector $\boldsymbol{c^*}$. The joint p.m.f for $\boldsymbol{C}$ and $\boldsymbol{c}^*$ can be computed as:

$$P(\boldsymbol{C}, \boldsymbol{c^*}|\boldsymbol{D}, \alpha, f) = P(\boldsymbol{c^*|\alpha})P(\boldsymbol{C}|\boldsymbol{c^*}, \boldsymbol{D}, f) \tag{2.10}$$

where $P(\boldsymbol{c^*|\alpha}) = \prod_{i=1}^{N} P(\lambda_i|\alpha)$, and $P(\boldsymbol{C}|\boldsymbol{c^*}, \boldsymbol{D}, f) = \prod_{i=1}^{N}\prod_{k=1}^{K} a_{ic_{ik}}$.

The p.m.f for $\boldsymbol{Z}$ can then be computed as:

$$P(\boldsymbol{Z|D}, \alpha, f) = \sum_{(\boldsymbol{c^*,C}):\phi(\boldsymbol{c^*,C})=\boldsymbol{Z}} P(\boldsymbol{C}, \boldsymbol{c^*}|\boldsymbol{D}, \alpha, f) \tag{2.11}$$

where ϕ is a many to one function that maps the connectivity matrix and ownership vector to the appropriate $\boldsymbol{Z}$ matrix.

CHAPTER 3

RESEARCH PLAN

We will propose a distribution for infinitely sparse matrices, incorporating distance information. Though Frazier and Blei have defined the distance dependent Indian buffet process (ddIBP), the distribution we are proposing will incorporate a measure of attraction similar to in the EPA distribution. We will perform simulations to explore and compare the properties of the proposed distribution and the ddIBP. For instance, we will explore expected row sums and column sums of random draws from the proposed distribution, expected value for matrices, expected value of rows given a predetermined set of previous rows, etc. Time permitting, we will also develop an MCMC method to make posterior inference, and explore the theoretical properties of and applications of the proposed distribution.

The p.m.f for $\boldsymbol{Z}$ in the ddIBP involves the enumeration of all possible sets of connectivity matrices and ownership vectors that generates the same $\boldsymbol{Z}$ matrix. This becomes infeasible very quickly as the dimensions of $\boldsymbol{Z}$ increases. Evaluating the p.m.f, therefore, cannot be done except for smaller dimensions of $\boldsymbol{Z}$, and typical Metropolis-Hastings algorithms based just on the likelihood and prior, cannot be conveniently implemented. We suspect that we can write a p.m.f. in closed form for the proposed distribution.

The new distribution we will propose will incorporate attraction information to determine how customers are connected by dishes. It will also preserve properties of the original IBP, such as expected number of dishes drawn by each customer, expected number of dishes drawn in total, and expected sum of the $\boldsymbol{Z}$ matrix.

CHAPTER 4

RESEARCH RESULTS

In this chapter, we will discuss the results of the research. We will discuss the sampling algorithm for the Attraction Indian buffet process (AIBP) and derive the p.m.f. of the distribution. Through a simulation study, we will also compare the properties of the IBP, AIBP, and ddIBP.

4.1 Notation for the AIBP

Before discussing the sampling algorithm and the p.m.f. for the AIBP, we will present the notation that will be used. In this chapter, the term "customer" will be used to refer to a row (or an observation) in a dataset; the term "dish" will likewise be used to refer to a column in a dataset. The terminology used here is consistent with that used in the IBP and ddIBP.

The order of customers in the dataset is not necessarily the order that the observations will be assigned dishes in implementation. We will call $\boldsymbol{\sigma} = (\sigma_1, .., \sigma_n)$ the *permutation* of the data. The permutation gives the sequence in which the n customers are assigned dishes, where the t^{th} customer assigned is σ_t. Note that $\boldsymbol{\sigma} = 1, ..., n$ when the customers are assigned in the order they appear in the dataset.

We will call $\lambda(i, j)$ the *similarity* between customers i and j. The function $\lambda(\cdot)$ maps the distance between customers to a measure of how "close" together the customers are. If d_{ij} represents the distance between customers i and j, then $\lambda(i, j) = f(d_{ij})$ should be a non-increasing function such that $\lambda(0) > 0$ and $\lambda(\infty) = 0$. The distance between customers should be non-negative. Examples of $\lambda(i, j) = f(d_{ij})$ include: (i) reciprocal similarity $f(d) = 1/d$, and (ii) exponential similarity $f(d) = exp(-d)$.

We will let x_i be the number of *new* dishes that customer i takes. We will set $y_{-i} = \sum\limits_{j=1}^{i-1} x_j$, the number of existing dishes before customer i draws new dishes, for $i = \{2, ..., N\}$. We will define $y_{-i} = 0$. Note that $g_i(x_i) = \dfrac{(\alpha/i)^{x_i}\exp(-\alpha/i)}{x_i!}$, which is the probability mass function for a Poisson distribution with parameter α/i and argument x_i. We will let the number of (non-zero) columns in $\boldsymbol{Z}$ be K. $m_{-i,k}$ = the number of customers that took dish k before customer customer i samples dishes. We will let $m_{-i} = \sum\limits_{k=1}^{K} m_{-i,k}$ be the total number of dishes taken before customer i samples any dishes. We will assign

$$h_{i,k} = \frac{\sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}}{\sum\limits_{k=1}^{y_i} \sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}},$$

the similarity component of the weight given to sampling dish k for customer i. The probability of customer i drawing dish k will be $p_{i,k} = h_{i,k}\dfrac{m_{-i}}{i}$. Note that when λ is a constant, $p_{i,k}$ reduces to $m_{-i,k}$ as in the IBP. We will define $H_N = \sum\limits_{i=1}^{N} \dfrac{1}{i}$ to be the harmonic number.

4.2 Sampling Algorithm for the AIBP

We will now introduce the sampling algorithm for the AIBP for a given permutation $\boldsymbol{\sigma}$. The sampling algorithm is similar to that of the IBP, with minor changes to the sampling of previously sampled dishes. We will use $\boldsymbol{Z}$ to denote a realization from the AIBP with parameter α, where α is a mass parameter that governs the number of dishes that will be drawn. The larger α is, the larger the number of total dishes drawn (K) will be.

To obtain a realization $\boldsymbol{Z}$ from an AIBP(α), the first customer draws an $x_1 \sim$ Poisson(α) number of dishes. This is indicated by setting $z_{1,1:x_1}$ to 1. For each subsequent customer i, if there exists any previously sampled dishes (i.e. if $y_{-i} > 0$), customer i samples dish k with probability $p_{i,k}$, for $k \in \{1, ..., y_{-i}\}$. The customer then samples an additional $x_i \sim$ Poisson(α/i) number of new dishes.

4.3 Probability Mass Function

Based on the sampling algorithm above, we can write down the p.m.f. for the proposed distribution as a product of probabilities for each customer taking each dish. Here we will define one more quantity: $\prod\limits_{k=1}^{y_{-i}} (p_{i,k})^{z_{i,k}}(1-p_{i,k})^{1-z_{i,k}} = 1$ if $y_{-i} = 0$.

$$
\begin{aligned}
\mathrm{P}(\boldsymbol{Z}|\boldsymbol{\sigma}) =& \prod_{i=1}^{N} \left\{ g_i(x_i) \prod_{k=1}^{y_{-i}} (p_{i,k})^{z_{i,k}}(1-p_{i,k})^{1-z_{i,k}} \right\} \\
=& \prod_{i=1}^{N} \left\{ \frac{(\alpha/i)^{x_i} \exp(-\alpha/i)}{x_i!} \prod_{k=1}^{y_{-i}} \frac{(h_{i,k}m{-}i)^{z_{i,k}}(i - h_{i,k}m_{-i,k})^{1-z_{i,k}}}{i} \right\} \\
=& \frac{\alpha^{\sum\limits_{i=1}^{N} x_i} \exp(-\alpha H_N)}{\prod\limits_{i=1}^{N} x_i!} \left(\prod_{i=1}^{N} i^{-x_i} \right) \times \\
& \left(\prod_{i=1}^{N} \prod_{k=1}^{y_{-i}} i^{-1} \left(\frac{\sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}}{\sum\limits_{k=1}^{y_i} \sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}} m_{-i} \right)^{z_{i,k}} \left(i - \frac{\sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}}{\sum\limits_{k=1}^{y_i} \sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}} m_{-i} \right)^{1-z_{i,k}} \right) \\
\\
=& \frac{\alpha^K \exp(-\alpha H_N)}{\prod\limits_{i=1}^{N} x_i!} \times \\
& \left(\prod_{i=2}^{N} i^{-(x_i + y_{-i})} \prod_{k=1}^{y_{-i}} \frac{\left(\sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k} m_{-i} \right)^{z_{i,k}} \left(i - \sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k} m_{-i} \right)^{1-z_{i,k}}}{\sum\limits_{k=1}^{y_{-i}} \sum\limits_{j=1}^{i-1} \lambda(\sigma_j, \sigma_i) z_{j,k}} \right)
\end{aligned}
$$

4.4 Properties of the AIBP

In the literature review, it was shown how each of the mentioned distributions that made use of pairwise distance information could be reduced to their original distributions (CRP and IBP). The AIBP can likewise be reduced to the IBP when the similarity function is constant (i.e. when f(d) = c, a constant).

One of the properties we wished to preserve from the IBP was that the expected number of dishes taken by each customer would be α. The algorithm prescribed above allocates dishes to customer based on their proximity to other customers. That is if two customers are close together, they are more likely to share the same dishes. The number of new dishes sampled by each customer in the AIBP is the same as that in the IBP (and is equal to α/i), by construction. So it remains to show that the expected number of previously-sampled dishes r_i sampled by customer i in the AIBP is the same as that in the IBP. In the IBP, the expected row sum for the first row is α, which is the same as in the AIBP. So the expected row sum is the same for the first row. For subsequent rows, the expected value of r_i in the IBP is $\mathrm{E}\left[\sum_{k=1}^{y_{-i}} \frac{m_{-i,k}}{i}\right] = \mathrm{E}\left[\frac{m_{-i}}{i}\right] = \frac{\alpha(i-1)}{i}$. For the AIBP, the expected value of r_i is:

$$E\left[\sum_{k=1}^{y_{-i}} h_{i,k}\frac{m_{-i}}{i}\right] = E\left[\sum_{k=1}^{y_{-i}} \frac{\sum_{j=1}^{i-1}\lambda(\sigma_j,\sigma_i)z_{j,k}}{\sum_{k=1}^{y_i}\sum_{j=1}^{i-1}\lambda(\sigma_j,\sigma_i)z_{j,k}} \frac{m_{-i}}{i}\right] = E\left[\frac{m_{-i}}{i}\right] = \frac{\alpha(i-1)}{i}$$

Therefore, the expected row sums are the same for the AIBP and IBP. Note also that since the mechanism for drawing new dishes is the same as that for the IBP, the expected total number of dishes drawn, which is the total number of non-zero columns in $\boldsymbol{Z}$ is $\alpha/1 + \alpha/2 + ... + \alpha/N = \alpha H_N$, as in the IBP. The implication is that the effective dimensions of the IBP are preserved in the AIBP, but the assignment of previously-sampled dishes to customers is altered by and redistributed according to distance information.

4.5 Simulation Study

To better understand the behavior of the AIBP, a simulation study was conducted to compare the IBP, AIBP, and ddIBP. We are particularly interested in three questions: (1) Do customers that are "close" together tend to share similar dishes? (2) Do customers that are "far" away from one another tend to not share dishes? (3) Given a particular arrangement for the first arbitrary number of customers, how do subsequent customers choose dishes? [1] Through the simulation study, we were able to confirm that in simulation, the AIBP reduces to the IBP when the similarity function is set to a constant (see Figure 4.1).

Now we will examine the behavior of the AIBP using the distance matrix given in Table 4.1. This matrix is of particular interest because two pairs of customers (1 & 3, and 2 & 5) are close together. All other customer pairings are distant. One customer, customer 4, is far from every other customer. With these properties, this matrix may provide some interesting insights into the behavior of the AIBP. The similarity function used is the exponential function.

Figure 4.2 shows the expected value of the IBP, AIBP, and ddIBP simulated with $\alpha = 1.5$, and the distance matrix D in Table 4.1. The three distributions are clearly different. In the AIBP (middle), it is more probable for the 3^{rd} customer to take dish 1, than it is in the IBP. This is reasonable because the first customer tends to take the first dish frequently, and the

[1] The simulation study can be accessed by installing and loading the "shiny" package in R, then running the line:
> runGitHub('shinyTest','luiarthur')

The code for the simulation study can be viewed at:
https://github.com/luiarthur/shinyTest

A screen shot of the application used for the simulation study can be found in the appendix.

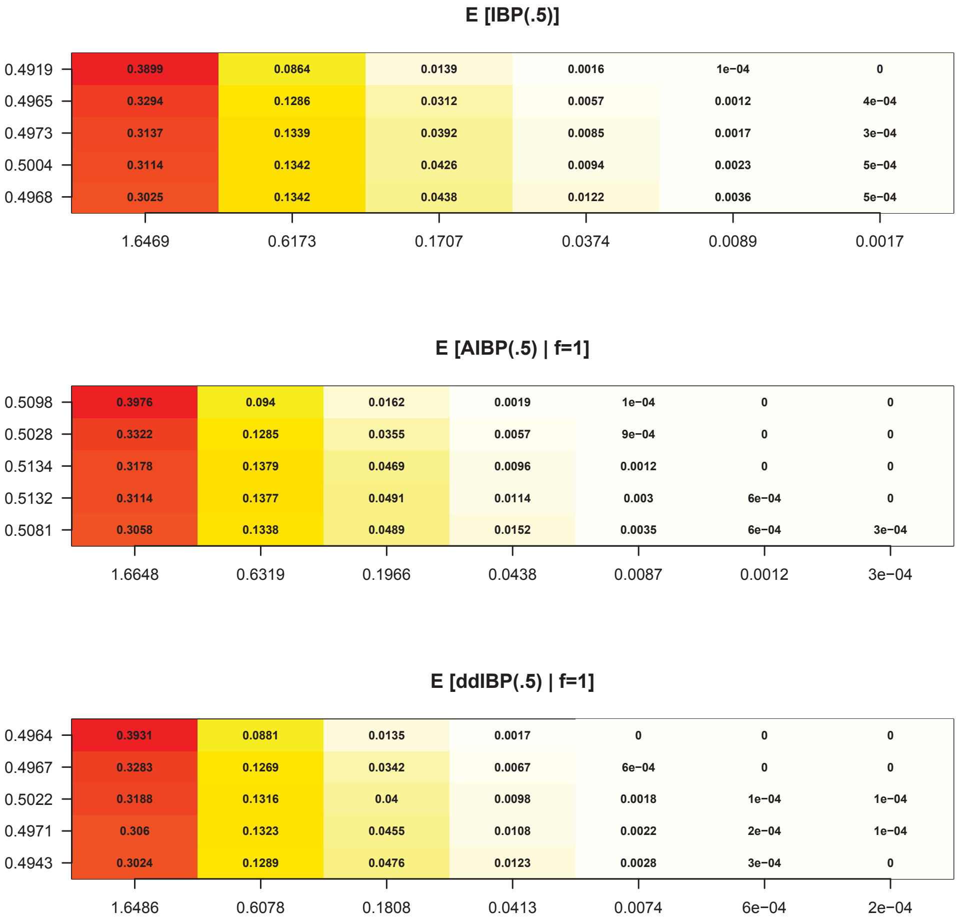


Figure 4.1: The expected values of draws from the IBP were computed via simulation. Ten thousand realizations were drawn from the IBP with $\alpha = .5$. The ten thousand matrices were then summed across and divided by the number of realizations (10,000). Because draws from the IBP do not always have the same dimensions, each of the matrices were first padded with zeros to make matrix summation conformable. Expected values were computed in a similar way for the AIBP and ddIBP. In simulation, we have shown that the AIBP and ddIBP reduce to the IBP when the similarity function is set to a constant value. E[ncol] is the expected number of columns for a given distribution. Values in the cells of the matrices are the probabilities of those cells taking the value '1' as opposed to 0. The colors are an additional indication of the probabilities, red is used for cells with higher probability; white is used for cells with lower probability. The numbers to the left of the grids are the expected row sums. The numbers at the bottom of the grids are the expected column sums.

$$\begin{pmatrix} 0 & 9 & 1 & 9 & 9 \\ 9 & 0 & 9 & 9 & 1 \\ 1 & 9 & 0 & 9 & 9 \\ 9 & 9 & 9 & 0 & 9 \\ 9 & 1 & 9 & 9 & 0 \end{pmatrix}$$

Table 4.1: Distance Matrix used in this simulation study.

3^{rd} customer is close to the first. In the ddIBP, customer 3 is more likely to take dish 1 than any other customer. This is the main difference between the behavior of the AIBP and the ddIBP. The ddIBP takes into account the distance information of each customer to every other customer. In the AIBP, the distance information for only customers assigned prior to the current customer is being made use of. This is neither a desireable nor an undesireable property, but is merely a property that distinguishes the AIBP from the ddIBP.

Figure 4.3 shows the expected values of the IBP, AIBP, and ddIBP, when the first two rows are predetermined. In this case, the first two columns of the first row were set to '1'; and the third and fourth columns of the second row were set to '1'. It is clear that the distance information is respected in both the ddIBP and the AIBP. It appears that in the AIBP, customers that are close to one another tend to share the same dishes. Customer 3 is close to customer 1, and consequently takes dishes 1 and 2 more frequently than he would in the IBP. Customer 4 is far away from all other customers. Or more accurately, he is equidistant from all other customers. So distance information does not have an effect on him. He takes dishes with probabiltiy proportional to their popularity, and not proportional to his proximity to customers that have taken those dishes. Customer 5 tends to take dishes 3 and 4 because he is close to customer 2, who has taken those dishes. In the ddIBP, customers that are far apart tend to not share dishes. This is observed by noticing that instead of

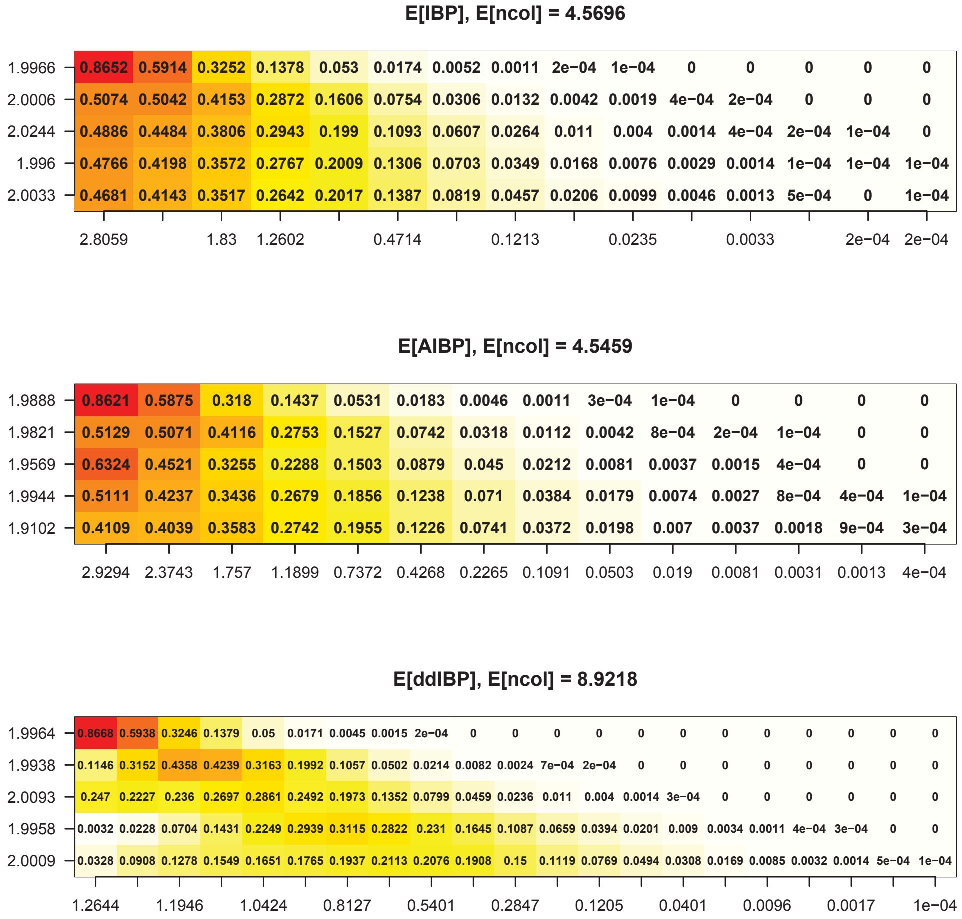


Figure 4.2: Expected value of the IBP, AIBP, and ddIBP, with $\alpha = 2$, and distance matrix D as shown in Table 4.1.

taking previously sampled dishes, customer 4 tends to sample more new dishes, because he is far from every other customer.

4.6 Comparisons between the AIBP & ddIBP

One obvious difference between the AIBP and ddIBP is that the pmf for any given binary matrix can be evaluated for the AIBP, while the pmf cannot be explicitly computed for the ddIBP. For both the AIBP and ddIBP, expected row sums for a given parameter α is α, which

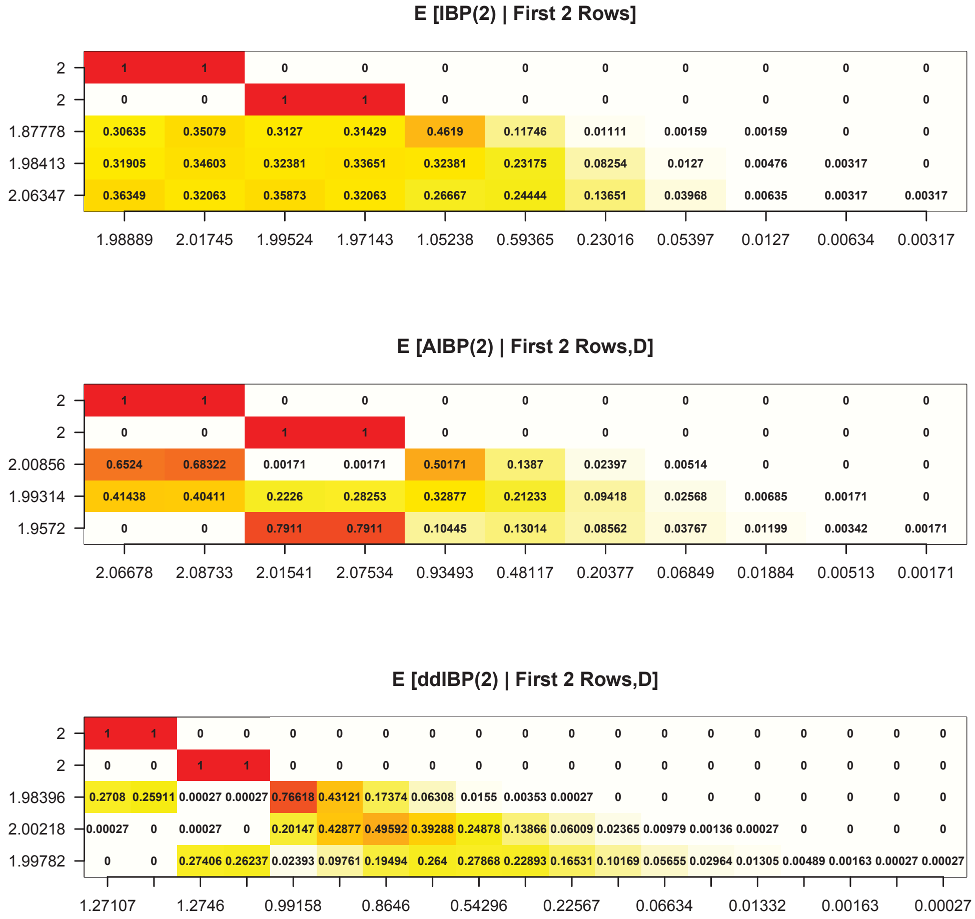


Figure 4.3: Expected value of the IBP, AIBP, and ddIBP with $\alpha = 2$, and distance matrix D as shown in Table 4.1, with the first two rows pre-set.

is also the expected row sum for the IBP(α). The expected number of non-zero columns in the AIBP is αH_N, which is also the expected number of non-zero columns in the IBP. This is because in the AIBP, the process for each "customer" drawing new dishes is the same as in the IBP. This equality does not hold for the ddIBP. In general, expected column sums are not equal to the IBP, for netiher the AIBP and ddIBP. However, the expected matrix sums for both the AIBP and ddIBP are the same as that of the IBP. Table 4.2 summarizes these findings. One implication of this result is that the AIBP preserves the dimensions of the IBP, but redistributes “dishes” to each customer based on proximity to other customers. The ddIBP does not preserve the dimensions of the IBP in this manner.

Comparison	AIBP	ddIBP
Explicit pmf	Yes	No
Expected non-zero columns equal to that of IBP	Yes	No
Expected row sums equal to that of IBP	Yes	Yes
Expected column sums equal to that of IBP	No	No
Expected matrix sum equal to that of IBP	Yes	Yes

Table 4.2: Comparisons of the AIBP to the ddIBP showing how what properties of the IBP they presrve.

CHAPTER 5

RESEARCH CONCLUSIONS & FUTURE WORK

The unique properties of the AIBP are that (1) it has an explicit p.m.f., and (2) it uses distance information to tilt the distribution of dish-taking while respecting the natural dimensions of the IBP. Having an explicit p.m.f. is certainly more frequently a modeling advantage. It is not clear whether preservation of dimensions is an advantage of the AIBP over the ddIBP. More study will be done to investigate the situations in which this property may be appropriate for a modeling choice.

In order to make use of the p.m.f. in MCMC to obtain posterior distributions, an appropriate proposal mechanism needs to be developed. The complexity occurs when the dimensions of the realizations from the IBP are changed from draw to draw. The proposal mechanism needs to account for added columns, removed columns, and changed values of the matrices drawn.

So far, we have not discussed a practical area of application for the IBP. But there may be appropriate and natural applications in the development Bayesian mixed models. The linear mixed model can be written as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}.$$

Traditionally, $\boldsymbol{Z}$ is a design matrix determined by researchers. The design matrix can consist of random intercepts for each subject in the dataset, and could also include random slopes. An interesting application of the IBP could be to model the design matrix $\boldsymbol{Z}$ with an IBP prior. Though it is usually predetermined, modeling $\boldsymbol{Z}$ may be a way of determining latent features (in this case, intercepts) that are generating the observations. If a subset of the observations are generating from common intercepts, modeling $\boldsymbol{Z}$ may be of value. Adding distance information using the AIBP may be suitable if we know that observations that are similar are more likely to share intercepts. However, no formal research has been conducted on this subject, and the application of the IBP in mixed models warrants further investigation.

CHAPTER 6

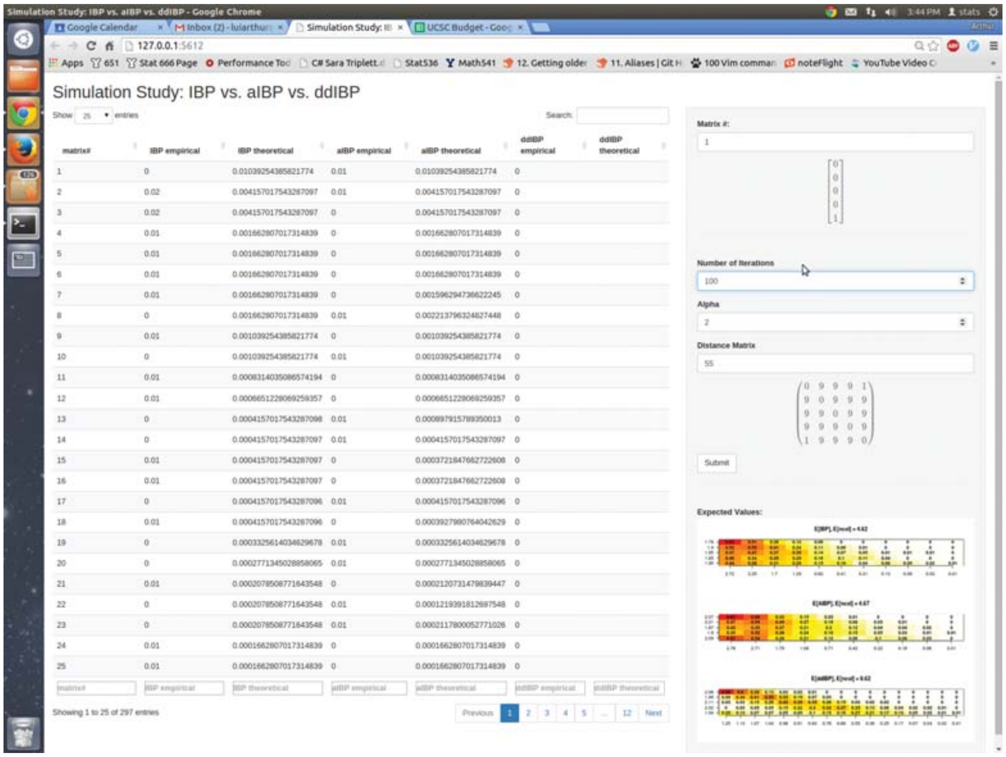
APPENDIX:

Figure 6.1 shows the *shiny* application created for the simulation study. The application can be launched by first installing the "shiny" package in R. Then library by typing:

```
> library(shiny)
> runGitHub("shinyTest","luiarthur")
```

The code for this project can be found at:

https://github.com/luiarthur/shinyTest



Simulation Study: IBP vs. aIBP vs. ddIBP

Show 25 entries

Search:

matrix#	IBP empirical	IBP theoretical	aIBP empirical	aIBP theoretical	ddIBP empirical	ddIBP theoretical
1	0	0.01039254385821774	0.01	0.01039254385821774	0	
2	0.02	0.004157017543287097	0.01	0.004157017543287097	0	
3	0.02	0.004157017543287097	0	0.004157017543287097	0	
4	0.01	0.001662807017314839	0	0.001662807017314839	0	
5	0.01	0.001662807017314839	0	0.001662807017314839	0	
6	0.01	0.001662807017314839	0	0.001662807017314839	0	
7	0.01	0.001662807017314839	0	0.001596294736622245	0	
8	0	0.001662807017314839	0.01	0.002213796324827448	0	
9	0.01	0.001039254385821774	0	0.001039254385821774	0	
10	0	0.001039254385821774	0.01	0.001039254385821774	0	
11	0.01	0.0008314035086574194	0	0.0008314035086574194	0	
12	0.01	0.0006651228069259357	0	0.0006651228069259357	0	
13	0	0.0004157017543287098	0.01	0.000897915789350013	0	
14	0	0.0004157017543287097	0.01	0.0004157017543287097	0	
15	0.01	0.0004157017543287097	0	0.0003721847662722608	0	
16	0.01	0.0004157017543287097	0	0.0003721847662722608	0	
17	0	0.0004157017543287096	0.01	0.0004157017543287096	0	
18	0.01	0.0004157017543287096	0	0.0003927980764042629	0	
19	0	0.0003325614034629678	0.01	0.0003325614034629678	0	
20	0	0.0002771345028858065	0.01	0.0002771345028858065	0	
21	0.01	0.0002078508771643548	0	0.0002120731479839447	0	
22	0	0.0002078508771643548	0.01	0.0001219391812697548	0	
23	0	0.0002078508771643548	0.01	0.0002117800052771026	0	
24	0.01	0.0001662807017314839	0	0.0001662807017314839	0	
25	0.01	0.0001662807017314839	0	0.0001662807017314839	0	

Showing 1 to 25 of 297 entries

Matrix #: 1

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Number of Iterations: 100

Alpha: 2

Distance Matrix: 55

$$\begin{pmatrix} 0 & 9 & 9 & 9 & 1 \\ 9 & 0 & 9 & 9 & 9 \\ 9 & 9 & 0 & 9 & 9 \\ 9 & 9 & 9 & 0 & 9 \\ 1 & 9 & 9 & 9 & 0 \end{pmatrix}$$

Expected Values:

Figure 6.1: Screen shot of the "shiny" application used for the simulation study.

BIBLIOGRAPHY

Blei, D. M., and Frazier, P. I. (2011), "Distance Dependent Chinese Restaurant Process," *Journal of Machine Learning Research*, 12, 2383–2410.

Dahl, D. B., Day, R., and Tsai, J. W. (2014), "Random Partition Distribution Indexed by Pairwise Information," .

Gershman, S. J., Frazier, P. I., and Blei, D. M. (2012), "Distance Dependent Infinite Latent Feature Models," *arXiv:1110.5454v2*.

Griffiths, T. L., and Ghahramani, Z. (2011), "The Indian Buffet Process: An Introduction and Review," *Journal of Machine Learning Research*, 12, 1185–1224.

Neal, R. M. (2000), "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, 9, 249–265.