Bayesian Analysis (0000) **00**, Number 0, pp. 1

The Attraction Indian Buffet Distribution

Richard L. Warr[*,‡], David B. Dahl[*], Jeremy M. Meyer[*], and Arthur Lui[†]

Abstract. We propose the attraction Indian buffet distribution (AIBD), a distribution for binary feature matrices influenced by pairwise similarity information. Binary feature matrices are used in Bayesian models to uncover latent variables (i.e., features) that explain observed data. The Indian buffet process (IBP) is a popular exchangeable prior distribution for latent feature matrices. In the presence of additional information, however, the exchangeability assumption is not reasonable or desirable. The AIBD can incorporate pairwise similarity information, yet it preserves many properties of the IBP, including the distribution of the total number of features. Thus, much of the interpretation and intuition that one has for the IBP directly carries over to the AIBD. A temperature parameter controls the degree to which the similarity information affects feature-sharing between observations. Unlike other nonexchangeable distributions for feature allocations, the probability mass function of the AIBD can be written explicitly and has a tractable normalizing constant, making posterior inference on hyperparameters straight-forward using standard MCMC methods. A novel posterior sampling algorithm is proposed for the IBP and the AIBD. We demonstrate the feasibility of the AIBD as a prior distribution in feature allocation models and compare the performance of competing methods in simulations and a real-data example.

Keywords: Bayesian nonparametric models, clustering, Chinese restaurant process, feature allocations, Indian buffet process, latent feature models.

1 Introduction

Two primary functions for data modeling are to relate observed data to each other and to future observations. These purposes of modeling imply that the data (both previously observed and yet observed) are, to some extent, related to each other. Thus, when modeling, we assume that the observed data are somehow interconnected and possess some information about future observations. These relationships are often complex and not easily captured in traditional models. Bayesian nonparametric latent feature models account for these complexities by allowing any number of features to connect observations to one another, without assuming a predetermined relationship structure.

One prior for Bayesian nonparametric latent feature models is the Indian buffet process (IBP) (Griffiths and Ghahramani, 2011). In a realization of the IBP, an observation may possesses zero, one, or any number of features possibly shared with the other observations. The total number of possible features is unbounded, and can theoretically account for any amount of complexity in the data. Under the Bayesian construct, the

[*] Brigham Young University, 223 TMCB, Provo, UT 84602
[†] UC Santa Cruz, 1156 High Street, Santa Cruz, CA 95064
[‡] Corresponding author: Richard L. Warr, warr@byu.edu

 DOI: 0000

IBP is used as a prior distribution for a feature allocation, and is updated with data (via a likelihood) which results in a posterior distribution of that feature allocation.

A major assumption of the IBP is that all observations are exchangeable. In other words, before the data are collected one item is indistinguishable from another. This assumption can be quite restrictive if, *a priori*, information about the observations is known. For example, the amount of trade between pairs of countries might be known, yet this information cannot be incorporated into a model which insists on an exchangeable feature allocation distribution.

To account for distance information between observations, Gershman et al. (2015) developed the distance dependent Indian buffet process (dd-IBP). This method allows a modeler to indicate, *a priori*, the distances between each pair of observations. In this paper we also propose a generalization of the IBP which incorporates pairwise distances into the feature allocation prior, namely the attraction Indian buffet distribution (AIBD). However, the AIBD retains a few desirable characteristics of the IBP which are lost with the dd-IBP. The first is that our method retains the same number of expected features as the IBP, whereas the dd-IBP changes the number of expected features, with respect to the IBP. The AIBD also has an explicit probability mass function (pmf) which readily allows for standard MCMC techniques on hyperparameters. Another property of the AIBD is that the expected number of shared features between two customers can increase or decease (in relation to the IBP). In the dd-IBP the expected number of shared features typically decreases as the distances are included. The methods associated with the AIBD are implemented in the package *aibd* available on CRAN. We feel, and demonstrate in detail, that the characteristics of our proposed method provide specific advantages over the dd-IBP.

The organization of the paper is as follows. In Section 2 we discuss some of the previous work for this methodology and establish the notation and models needed for this article. In Section 3 we present the AIBD pmf. Next, in Section 4, we investigate key properties of the AIBD and compare them to the dd-IBP. Then, in Section 5, we outline a new recipe for posterior simulation when using an IBP or AIBD prior. Section 6 is a description of a classification analysis of an Alzheimer's disease neuroimaging study (Dinov et al., 2009), and demonstrates the advantages of using an AIBD prior. We finish in Section 7 with a brief summary of this work.

2 Literature Review

In this section we discuss the primary literature needed for our proposed method and define notation used in this article. A short discussion of the Chinese restaurant process is included as an aide for those who might be familiar with random partition models, but are new to feature allocation models.

2.1 The Chinese Restaurant Process

Bayesian nonparametric models seek to capture latent structure in data. In clustering applications where each observation is assigned to a group to form a partition, the Chi-

nese restaurant process (CRP) serves as a prior distribution over all possible partitions. The CRP resembles a Chinese restaurant with an infinite number of tables, in which n customers enter one at a time. Each customer picks a table to sit at, favoring tables with more customers. The resulting assignment of customers to tables induces a partition of the customers. Thus, the CRP will create latent features that are exclusive. The CRP is exchangeable. Consequently, the probability of any two customers being in the same cluster is the same for all pairs of customers.

However, the constraint that each customer has an equal chance of being clustered with any other customer may not fully reflect existing (*a priori*) knowledge. Certain covariates like socioeconomic background, age, or other distances in time and space, will likely impact the clustering. Therefore, instead of constraining all datapoints to be equidistant at the start of the analysis, it could be useful to have an expert incorporate pairwise distances into the prior. Blei and Frazier (2011) developed the distance-dependent Chinese restaurant process (ddCRP) to facilitate these distances *a priori*. However, the ddCRP does not have an explicit probability mass function (pmf), which makes using standard Markov chain Monte Carlo (MCMC) techniques for posterior inference on hyperparameters difficult.

Dahl et al. (2017) proposed the Ewens-Pitman attraction distribution (EPA), which also allows pairwise distance information to be included in the CRP. This distribution has an explicit pmf with a tractable normalizing constant and pmf. This is ideal for using standard MCMC sampling methods. Like the dd-CRP, the EPA places more probability on partitions that group similar items. But unlike the ddCRP, the EPA does not change the distribution of the number of subsets; it only influences how the datapoints are clustered together within the class of partitions having the same number of subsets.

Similar to how the EPA incorporated distance information while preserving many of the qualities of the CRP, we propose a new distribution, the AIBD, that incorporates pairwise distance information in the IBP prior. Although the existing dd-IBP uses pairwise distances, we propose a distribution that preserves some properties and intuition for the IBP and has an explicit pmf with a tractable normalizing constant.

2.2 The Indian Buffet Process

A popular prior distribution for Bayesian nonparametric latent feature allocation models is the Indian buffet process (Griffiths and Ghahramani, 2006). The Indian buffet process (IBP) puts a prior distribution on feature allocations. The generative construct of the IBP can be thought of as an Indian buffet restaurant with a seemingly infinite number of dishes. A fixed number of customers enter the buffet one at a time to sample dishes. The first customer enters and takes a Poisson(α) number of unique dishes. After the first customer, the i^{th} customer samples each existing dish with probability m_k/i, where m_k is the number of customers who have previously sampled dish k. The i^{th} customer then takes Poisson(α/i) new dishes. Thus, popular dishes will tend to be taken more often by later customers and the number of new dishes to be sampled will diminish as more customers enter the restaurant.

The dishes taken by each customer can be encoded in a (binary) feature allocation matrix $\mathbf{Z}$ where rows and columns correspond to customers and dishes, respectively. In this matrix, $z_{i,k} = 1$ indicates that customer i took dish k. Likewise $z_{i,k} = 0$ indicates that customer i did not take dish k. $\mathbf{Z}$ also describes how the customers share features. For example, $z_{i,k} = z_{j,k} = 1$ indicates customers i and j both took (i.e. share) dish k. The dishes are analogous to latent features and thus customers who share more dishes are thought to share similar (unobserved) attributes. Although, technically, an infinite number of dishes are not sampled (which are represented as columns of zeros), these are generally removed from $\mathbf{Z}$.

Since the dishes are indistinguishable, the ordering of the columns in $\mathbf{Z}$ is irrelevant. As a result, any permutation of the columns in $\mathbf{Z}$ will correspond to the same feature allocation. Considering all column-permutations of $\boldsymbol{Z}$ that represent the same feature allocation is important. One way to map each $\boldsymbol{Z}$ to its unique feature allocation is to consider the equivalence class of matrices in left-ordered form. A left-ordered form (*lof*) matrix can be obtained by taking the binary number of each column (with the most significant digit in the first row) and then ordering the columns in descending order from left to right. A $\mathbf{Z}$ in left-ordered form will thus have a stair-like pattern, with the first 1 appearing in a column only when a new dish is taken. To take into account the indistinguishable columns, we add a combinatoric term to the probability mass functions in Equations (1) and (5). Thus a specific $\boldsymbol{Z}$ will refer to the class of all feature allocations that map to the same left-ordered form.

The expected number of sampled dishes per customer is the mass parameter α, a positive real number. We will denote N as the number of customers and K as the total number of dishes taken by at least one customer. That is, the matrix $\boldsymbol{Z}$ has N rows and K nonzero columns. Define x_i as the number of new dishes customer i takes, y_i as the number of sampled dishes before customer i, and $m_{-i,k}$ as the number of customers that took dish k before customer i. For convenience, let $H_N = \sum_i^N (1/i)$ be the N^{th} harmonic number. The IBP pmf is shown in Equation (1) and can be loosely divided into 3 pieces: the combinatorial term, the Poisson term, and the Bernoulli (binary) term. The cardinality of all possible non-zero binary columns of length N is $2^N - 1$, so $\prod_{h=1}^{2^N-1} K_h!$ iterates over the sample space of all distinct non-zero columns in $\mathbf{Z}$. Where K_h represents the number of columns for the h^{th} possible configuration. By following the constructive pattern, the IBP pmf can be expressed as

$$\mathrm{P}(\mathbf{Z}|\alpha) = \left[\frac{\prod_{i=1}^{N} x_i!}{\prod_{h=1}^{2^N-1} K_h!}\right] \frac{\alpha^K \exp\{-\alpha H_N\}}{\prod_{i=1}^{N} (i^{x_i}\, x_i!)} \prod_{i=2}^{N} \prod_{k=1}^{y_i} \left(\frac{m_{-i,k}}{i}\right)^{z_{i,k}} \left(1 - \frac{m_{-i,k}}{i}\right)^{1-z_{i,k}}. \tag{1}$$

Note that the $\prod_{i=1}^{n} x_i!$ term will cancel, but it is not removed so one can intuitively see the origin of the various parts of the pmf. In the case where no dishes have been sampled before customer i enters ($y_i = 0$), the result of the double product is defined to be 1.

The IBP prior has the property that customers are exchangeable, i.e. changing the order of the rows in $\mathbf{Z}$ has no impact on the probability of any given feature allocation.

As a result, the expected number of shared features for all customers is uniform. Therefore, on average, customers will share the same number of features. While this may be desirable in instances where nothing is known about the customers, additional *a priori* information may be relevant and should be used to influence how features are shared. For example, in the context of the Indian buffet restaurant, we may know that certain customers have similar dietary preferences before they walk in the restaurant. In other applications, time-based, spatial, or covariate dependencies can be used to create prior dependencies between data points. Instead of assuming customers are exchangeable before the analysis, it may be helpful to relax this assumption in light of additional information.

The Distance Dependent IBP

Gershman et al. (2015) proposed a generalization of the IBP, the distance dependent Indian buffet process (dd-IBP), which incorporates pairwise distance information. Compared to the IBP, this distribution, on average, inhibits feature-sharing as a function of distance; that is, the larger the distance, the less sharing occurs. The distance can be defined on any metric, allowing for flexibility in defining the closeness of two data points. Similar to the IBP, each customer selects a Poisson number of dishes. However, the dd-IBP works differently from the IBP in that for each dish, the customers share dishes based on customer connections. The probability that one customer connects with another decreases as the pairwise distance increases. The resulting feature allocation is then derived from an intricate web of connections between customers. The web of connections between items implies a feature allocation, and the probability rules for these connections implicitly define a probability distribution on the induced feature allocation. One challenge of this approach, however, is that posterior inference on hyperparameters is difficult or impossible since standard MCMC techniques require an explicit pmf with a tractable normalizing constant. The dd-IBP reduces to the IBP when the proximity matrix is a lower diagonal matrix of 1's. Since the AIBD will also incorporate pairwise distance information, we will compare properties of both the dd-IBP and AIBD in Section 4.

Recent Applications of and Other Work on the IBP

The value of the IBP can be seen in its repeated application in research problems, particularly, in the biological sciences. See, for example, Hai-son and Bar-Joseph (2011); Chen et al. (2013); Xu et al. (2013); Sengupta et al. (2014); Xu et al. (2015); Lee et al. (2015, 2016); Ni et al. (2018); Lui et al. (2020).

In terms of methodological extensions, other work has been done to relax the exchangeability constraint of the IBP in the literature. Williamson et al. (2010) proposed the dependent IBP, which introduces dependence through a hierarchical Gaussian process. Miller et al. (2012) proposed a generalization of the IBP, the phylogenetic Indian buffet process, that introduces dependencies between objects by conditioning on a dependency tree. This also reduces down to the IBP when all branches meet at the root. This method performs well for data with genealogical relationships and expresses

prior object similarity through a tree. The Indian buffet Hawkes process (Tan et al., 2018) extended the IBP to capture latent temporal dynamics by incorporating ideas from the Hawkes process. Williamson et al. (2020) presents a class of nonexchangeable dynamic models constructed by adapting the IBP. These models are tailored to data that are believed to be generated by latent features exhibiting temporal persistence. We focus our comparison on the dd-IBP since it, like our AIBD, introduces dependence through pairwise distances.

2.3 The Linear Gaussian Latent Feature Model (LGLFM)

The typical likelihood in the Bayesian nonparametric literature for latent feature models is the linear Gaussian latent feature model (LGLFM). Using similar notation as found in Griffiths and Ghahramani (2011), the LGLFM is defined as:

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A} + \varepsilon, \tag{2}$$

where $\boldsymbol{X}$ is an $N \times D$ matrix of N observations on D variables. $\boldsymbol{Z}$ is an $N \times K$ binary matrix of 0s and 1s and indicates which features are turned off or on for a specific observation (i.e., row of $\boldsymbol{X}$). $\boldsymbol{A}$ is a $K \times D$ matrix whose rows are the latent features and whose prior is a matrix Gaussian distribution, with probability density function

$$p(\boldsymbol{A}|\sigma_A) \propto \exp\left\{-\frac{1}{2\sigma_A^2}\text{trace}\left(\boldsymbol{A}^T\boldsymbol{A}\right)\right\}. \tag{3}$$

Finally, ε is an $N \times D$ matrix and represents the error term of the model; it also has a matrix Gaussian distribution similar to $\boldsymbol{A}$ but has different dimensions and its parameter is σ_X. Technically $\boldsymbol{Z}$ and $\boldsymbol{A}$ have an infinite number of columns and rows (respectively). However, only K columns of $\boldsymbol{Z}$ are non-zero. Thus the zero columns of $\boldsymbol{Z}$ are discarded along with the associated rows of $\boldsymbol{A}$ and we treat those matrices as if they have a finite number of rows and columns (see Griffiths and Ghahramani (2005) or Griffiths and Ghahramani (2011) for more details). If $\boldsymbol{A}$ is integrated out of the model, the collapsed likelihood is:

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{Z},\sigma_X,\sigma_A) \propto & \frac{1}{\sigma_X^{ND-KD}\sigma_A^{KD}\left|\boldsymbol{Z}^T\boldsymbol{Z} + \frac{\sigma_X^2}{\sigma_A^2}\boldsymbol{I}\right|^{D/2}} \times \\ & \exp\left\{-\frac{1}{2\sigma_X^2}\text{trace}\left(\boldsymbol{X}^T\left[\boldsymbol{I} - \boldsymbol{Z}\left(\boldsymbol{Z}^T\boldsymbol{Z} + \frac{\sigma_X^2}{\sigma_A^2}\boldsymbol{I}\right)^{-1}\boldsymbol{Z}^T\right]\boldsymbol{X}\right)\right\}. \end{aligned} \tag{4}$$

We use this collapsed likelihood in the posterior inference section with various feature allocation priors on $\boldsymbol{Z}$ in Section 5.

3 The Attraction Indian Buffet Distribution

We propose a generalization of the IBP, the attraction Indian buffet distribution (AIBD). We describe how we obtain this distribution by modifying the generative model of the

IBP to include distance information between customers. Incorporating existing distance information about the customers, in turn, influences how the dishes are shared. We then show the nonexchangeable probability mass function and compare it to the IBP.

Distance information can be stored in a symmetric $N \times N$ pairwise distances matrix. The distance d_{ij} between customers i and j is located in the i^{th} row and j^{th} column. We transform these distances to similarity values, where 0 indicates negligible similarity and larger values indicate a greater similarity between customers. Various transformations can appropriately map distance to similarity and a temperature parameter τ is introduced to accentuate the effect of these distances. In general, we require: i. the transformation function $f(d_{ij}, \tau)$ to be a non-increasing function in d_{ij} for fixed τ, ii. $f(\tau_1, d_1)f(\tau_2, d_2) \leq f(\tau_2, d_1)f(\tau_1, d_2)$ for $d_1 \leq d_2$ and $\tau_1 \leq \tau_2$, and iii. when $\tau = 0$, it must return a constant in the interval $(0, \infty)$. The tranformation used throughout the paper is the exponential decay function: $f(\tau, d_{ij}) = \exp(-\tau d_{ij})$. The result of the element-wise transformations of the distance matrix is a similarity matrix. The AIBD uses the similarity matrix $\mathbf{\Lambda}$ to incorporate dependence between customers. A temperature parameter is also used in the dd-IBP and a few functions to transform distances to proximities are suggested in Gershman et al. (2015). It is worth mentioning that the dd-IBP does not require a symmetric distance matrix.

Due to non-exchangeability, the AIBD is conditioned on a permutation parameter $\boldsymbol{\rho}$, which is any permutation vector of the integers 1 to N. This controls the order in which customers arrive and allows us to characterize temporal or spatial dependence *a priori*. In many cases, however, the data has no natural ordering. That is, it may not make sense to say the data depends on the order it was observed or recorded. For this reason, the permutation parameter is sometimes averaged out of the model by Monte Carlo integration or enumeration.

Because we desire to preserve many of the properties of the IBP, the AIBD has a generative model very similar to the IBP. Using the same restaurant analogy as the IBP, the AIBD can also be thought of as an Indian buffet restaurant where customers enter one at a time. Like the IBP, the first customer takes a Poisson(α) number of dishes and the i^{th} customer takes Poisson(α/i) new dishes. However, instead of sampling existing dishes with probability proportional to the number of customers who have already sampled the dish, the AIBD also uses pairwise similarity information. The i^{th} customer gets existing dishes with probability equal to the sum of similarities of individuals who have that dish, divided by the sum of the total similarity with all previous individuals, all multiplied by $(i-1)/i$. When all the pairwise similarities are the same, the probability of sampling existing dishes reduces to that in the IBP. Thus, the IBP can be thought of as a special case of the AIBD when all the pairwise similarity components are identical.

By following the constructive process described above, the pmf of the AIBD can be obtained, as shown in Equation (5). The AIBD is a distribution over feature allocations. So, the support is over all binary matrices with N rows and only non-zero columns. Since the ordering of features does not matter, this probability mass function returns the probability of all feature allocations that are equivalent to the supplied $\boldsymbol{Z}$. The AIBD uses a pairwise distance matrix $\boldsymbol{D}$ and is conditioned on the permutation vector $\boldsymbol{\rho}$ of

the integers 1 to N. The parameters and notation carry the same meaning as they do in the IBP pmf in Equation (1). The pmf of the AIBD is

$$\mathrm{P}(\boldsymbol{Z}|\alpha,\boldsymbol{\rho},\tau) = \frac{\prod_{i=1}^{N} x_i!}{\prod_{h=1}^{2^N-1} K_h!} \cdot \frac{\alpha^K \exp\{-\alpha H_N\}}{\prod_{i=1}^{N}(i^{x_i}\, x_i!)} \prod_{i=2}^{N} \prod_{k=1}^{y_i} \left(\frac{h_{ik}(\tau)\cdot(i-1)}{i}\right)^{z_{i,k}} \left(1 - \frac{h_{ik}(\tau)\cdot(i-1)}{i}\right)^{1-z_{i,k}}. \tag{5}$$

The similarity component $h_{ik}(\tau)$ is defined as

$$h_{ik} = \frac{\sum_{j=1}^{i-1} f(\tau, d_{\rho_j,\rho_i}) \cdot z_{j,k}}{\sum_{j=1}^{i-1} f(\tau, d_{\rho_j,\rho_i})}. \tag{6}$$

The term d_{ρ_j,ρ_i} corresponds to the distance between the i^{th} and j^{th} individuals in a given permutation $\boldsymbol{\rho}$. As mentioned before, we use the function $f(\tau, d_{ij}) = \exp(-\tau d_{ij})$ to transformation distances to similarities, although there are many other possible functions that could perform that mapping.

The IBP and AIBD priors have the same support, and the probability mass functions are fairly similar. The key differences are the probabilities defined in the double product of Equations 1 and 5. The IBP has a customer sample a dish proportional to the number of times it has been taken. Customers in the AIBD also sample popular dishes more frequently, but the probability is also dependent on similarity information.

4 Properties of the AIBD

In this section, we explore some of the properties of the AIBD and compare them to the IBP and dd-IBP. A distribution on possible $\mathbf{Z}$'s implies a distribution on the number of non-zero columns. The distribution of the number of non-zero columns in the AIBD is the same as that in the IBP because, in the constructive model, the distance information is not used to determine the number of dishes. Thus, the distribution of features is invariant to similarity information included in the AIBD. We will compare this result by simulation to the dd-IBP, where the distribution of the number of features changes with the temperature parameter and the distance information. We will also compare how the features are shared between customers as a function of temperature for both the AIBD and dd-IBP. Thus, we will proceed by focusing on the total number of features and number of shared features for $\boldsymbol{Z}$.

4.1 Distribution of the Number of Features

In the IBP and AIBD, the distribution of the number of features T (i.e. number of non-zero columns in the $\mathbf{Z}$ matrix) can be explicitly characterized. Since a new column in $\mathbf{Z}$ is generated when a customer samples a new dish, the total of the number of features is

equal to the sum of the number of new dishes each customer takes. From the generative model of the IBP and AIBD, let the number of new dishes that the i^{th} customer takes be $X_i = \text{Poisson}(\alpha/i)$. Since the Poisson draws are independent between customers, the total number of features, $T = \sum_i X_i$ is distributed

$$T \overset{\text{d}}{=} T_{IBP} \overset{\text{d}}{=} T_{AIBD} \sim \text{Poisson}(\alpha H_N), \tag{7}$$

where H_N is the N^{th} harmonic number. This distribution is identical for the IBP and AIBD because the similarity information present in the AIBD is only used to determine how existing features are shared. No new dishes or columns are generated based on the distance information. Note that this is also invariant to the permutation parameter $\boldsymbol{\rho}$. The distribution of the number of features can only be changed by adjusting the mass parameter α or changing the number of customers N.

The generative model for the dd-IBP, however, is different in that the proximity, like the AIBD's similarity information, changes the total number of features. The dd-IBP uses a proximity matrix $\boldsymbol{P}$ to capture pairwise distances *a priori*. Using the dd-IBP generative model in Gershman et al. (2015), the number of new features for customer i is $X_i \sim \text{Poisson}(\alpha/h_i)$. Thus,

$$T_{dd\text{-}IBP} = \sum_{i=1}^{N} X_i \sim \text{Poisson}\left(\alpha \sum_{i=1}^{N} \frac{1}{h_i}\right), \tag{8}$$

where $h_i = \sum_{j=1}^{N} \boldsymbol{P}_{ij}$ and $\boldsymbol{P}_{ij}$ corresponds to the proximity measure between customers i and j. Thus, h_i is the sum of the i^{th} row in the dd-IBP proximity matrix. The proximity matrix in the dd-IBP differs slightly from the similarity matrix $\boldsymbol{\Lambda}$ in the AIBD. The dd-IBP requires self-proximity to be 1 and infinite distances to be mapped to a proximity of 0. Thus, only monotonic transformations that map distance values from $[0, \infty)$ to $[0, 1]$ can be used. We will employ the same transformation mentioned earlier, $f(\tau, d_{ij}) = \exp(-\tau d_{ij})$, where d_{ij} corresponds to the pairwise distance matrix used for the AIBD. The proximity matrix is called *sequential* if when $i < j$, $\boldsymbol{P}_{ij} = 0$; that is, when the proximity matrix is lower-diagonal. This allows current customers to only inherit dishes from previous customers. For the dd-IBP to simplify to the IBP, one condition is that $\boldsymbol{P}$ must be sequential, analogous to how the IBP restaurant analogy only allows customers to enter one at a time.

The dd-IBP simplifies to the IBP when the proximity matrix is a lower-diagonal matrix of 1's. When this happens, h_i in Equation (8) is equal to i, so $T_{dd\text{-}IBP}$ and T_{IBP} have the same distribution. A comparison of an AIBD similarity matrix using the natural permutation (integers 1 through N in ascending order) and a sequential dd-IBP proximity matrix is shown in Table 1. Both matrices were generated from the USArrests dataset in R. We selected the states New Hampshire, Iowa, Wisconsin, California, and Nevada respectively. We then calculated the pairwise Euclidean distances of the 5 states after centering and scaling the covariates. For the AIBD, we put the pairwise distances in a 5x5 matrix $\boldsymbol{D}$ as described in Section 3. Finally, to get the similarity or proximity, we applied the transformation $\exp(-\tau d_{ij})$ to each distance element.

AIBD Similarity Matrix ($\tau = 1$)

1	2	3	4	5
1.00	0.89	0.51	0.02	0.02
0.89	1.00	0.55	0.03	0.02
0.51	0.55	1.00	0.04	0.03
0.02	0.03	0.04	1.00	0.36
0.02	0.02	0.03	0.36	1.00

dd-IBP Proximity Matrix ($\tau = 1$)

1	2	3	4	5
1.00	0.00	0.00	0.00	0.00
0.89	1.00	0.00	0.00	0.00
0.51	0.55	1.00	0.00	0.00
0.02	0.03	0.04	1.00	0.00
0.02	0.02	0.03	0.36	1.00

Table 1: Similarity matrix $\mathbf{\Lambda}$ and sequential proximity matrix $\boldsymbol{P}$ at a fixed temperature τ using the natural permutation. Although both have ones on the diagonal, it is only required for the dd-IBP's proximity matrix $\boldsymbol{P}$.

After fixing $\alpha = 1, N = 5$, and using the permutation of increasing natural numbers 1 through N, we obtain the probability distribution of the number of features as shown in Figure 1. Recall that both T_{AIBD} and T_{IBP} have the same distribution as T, whereas the number of non-zero columns in the dd-IBP, $T_{dd\text{-}IBP}$, varies by temperature and distance information. The distributions of T and $T_{dd\text{-}IBP}$ were given in Equations (7) and (8). Figure 1 illustrates that, for this proximity matrix, the dd-IBP has a higher number of expected features than the IBP. As the temperature increases, $T_{dd\text{-}IBP}$ has a limiting distribution of a Poisson(αN), and values on the off diagonal of the dd-IBP's proximity matrix approach zero. Thus, the proximity matrix $\boldsymbol{P}$ approaches an identity matrix, and $\alpha \sum_i^n (1/h_i) \to \alpha N$ as $\tau \to \infty$. In this case, the N customers do not share features and individually sample α dishes, on average. On the other hand, the AIBD preserves the same distribution of the number of features as the IBP, regardless of temperature, and is only affected by the mass parameter. This is an important attribute of the AIBD distribution, because as the number of features increases so does the computational complexity of inference. The AIBD also encourages more feature sharing than the dd-IBP, as discussed in the next section.

4.2 Expected Number of Shared Features

One consequence of exchangeability in the IBP is that the expected number of shared features is identical for all customer pairs. This is not a desirable property when, *a priori*, one knows that a pair of customers are more alike when compared to another customer. The AIBD is able to include this information; which has the desired effect of changing the expected number of shared features for a pair of customers, while the average overall feature sharing in the restaurant remains the same as with the IBP.

In the AIBD, customers that are closer in distance tend to share more features. The degree to which customers share features can be adjusted by the temperature parameter. An example is shown in the plot of Figure 2. The plot was obtained by fixing $\alpha = 1$ and by using pairwise similarity information found in Table 1. We fixed α because changing the mass parameter only scales the y-axis in the plot of Figure 2. The lines indicate the expected number of shared features for a pair of customers. In this example, the customers are analogous to the individual states.

When the temperature is zero, the AIBD reduces to the IBP, and so all customer pairs have the same expected number of shared features. Due to variability between

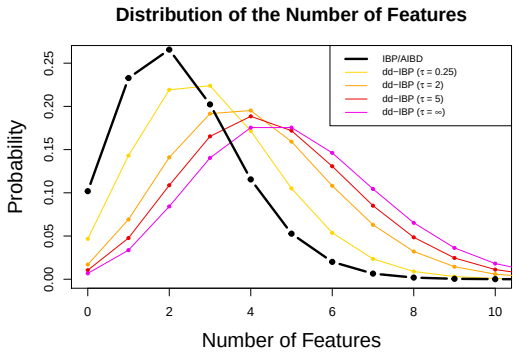
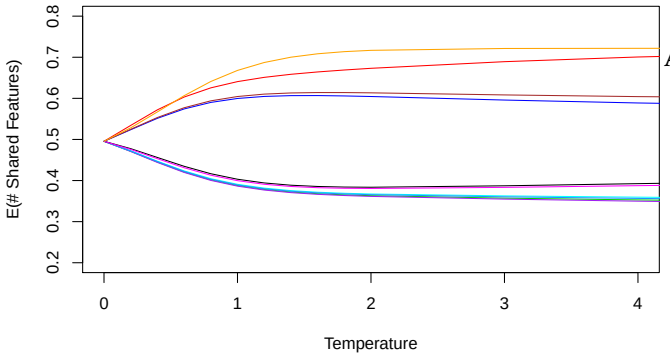


Figure 1: The distribution of T_{AIBD} and T_{IBP} is displayed as the bold black line. The distribution of $T_{dd\text{-}IBP}$ is displayed in the narrow colored lines for various temperatures. This figure shows that the distribution of the number of features for the dd-IBP (using the proximity matrix in Table 1 and when $\tau > 0$) is stochastically greater than the number of features for the IBP and the AIBD.



AIBD Similarity Matrix ($\tau = 1$)

1	2	3	4	5
1.00	0.89	0.51	0.02	0.02
0.89	1.00	0.55	0.03	0.02
0.51	0.55	1.00	0.04	0.03
0.02	0.03	0.04	1.00	0.36
0.02	0.02	0.03	0.36	1.00

Figure 2: The expected number of shared features as a function of temperature in the AIBD (averaged over all permutations) with corresponding similarity matrix when $\tau = 1$. This figure shows that the AIBD adjusts feature sharing for pairs of customers given the similarity matrix; the IBP would have all customers share on average 0.5 features.

permutations, each line in Figure 2 was calculated by averaging the expected number of shared features across all possible $N! = 120$ permutations. In other words, $\boldsymbol{\rho}$ was integrated out after placing a uniform prior on $\boldsymbol{\rho}$. Since N is small, we enumerated across all permutations up to 7 features, which accounted for 99.4% of the probability mass.

A similarity matrix from a fixed temperature is shown to the right of the temperature plot in Figure 2. As expected, customers with higher similarity tend to share more features. However, even though customers 1 and 2 have the highest similarity, customers 4 and 5 tend to share more features. One possible reason for this behavior is that customers 4 and 5 are more dissimilar to all other customers than customers 1 and 2 are.

We now examine the effect of increasing N on the expected number of shared features per customer in the AIBD. We will do this through plots similar to Figure 2. Additionally, we use plots for both the AIBD and dd-IBP to compare how they each influence feature sharing. We use sample sizes $N = 5$ and $N = 50$ to demonstrate the effect of increasing N. Since it is computationally infeasible to enumerate 50! permutations, we create the plots using Monte Carlo estimation.

To sample from the AIBD with a greater number of customers, we used pairwise distance information from all $N = 50$ states in the USArrests dataset in R. This was done using the same method as for $N = 5$. Thus, for a fixed temperature, the similarity matrix used when $N = 5$ is a matrix partition of the similarity matrix used when $N = 50$. The simulation results are shown in the top row of Figure 3.

Note that in the IBP, which corresponds to the AIBD with $\tau = 0$, the expected number of shared features between all customer pairs is $\alpha/2$. For any N, due to exchangeability, the number of shared features is the same for all customer pairs. Recall that the first customer samples $X_1 \sim \text{Poisson}(\alpha)$ dishes. The second customer will take each dish sampled by customer 1 with probability 1/2. Thus the total number of shared features between customers 1 and 2, is $X_{1,2} \sim \text{Binomial}(X_1, 1/2)$. By the law of total expectation, $E(X_{1,2}) = E(E(X_{1,2}|X_1)) = E(X_1/2) = \alpha/2$. Since any permutation of customers results in the same probability distribution for the IBP, the expected number of shared features for the first and second customer is the same for other pairs. While the AIBD is not exchangeable, it can be shown by simulation that across any temperature τ and sample size N, the expected number of shared features averaged across all pairs is $\alpha/2$. Thus, the AIBD allows each pair to share features differently; but across all pairs, the mean of the expected number of shared features is the same as the IBP.

For the AIBD, as customers are added to the process (i.e., as N grows) the behavior of the average feature sharing changes, as seen in the top two plots of Figure 3. For high temperatures, a customer pair can share, on average, more or less than when there are more customers in the restaurant. At $\tau = 5$ and $N = 5$, the expected number of shared features for the 10 pairs range from 0.35 to 0.73, when $N = 50$ this range increases by roughly 60% for those same pairs. Thus increasing τ and N allows more disparity between the average feature sharing of customer pairs.

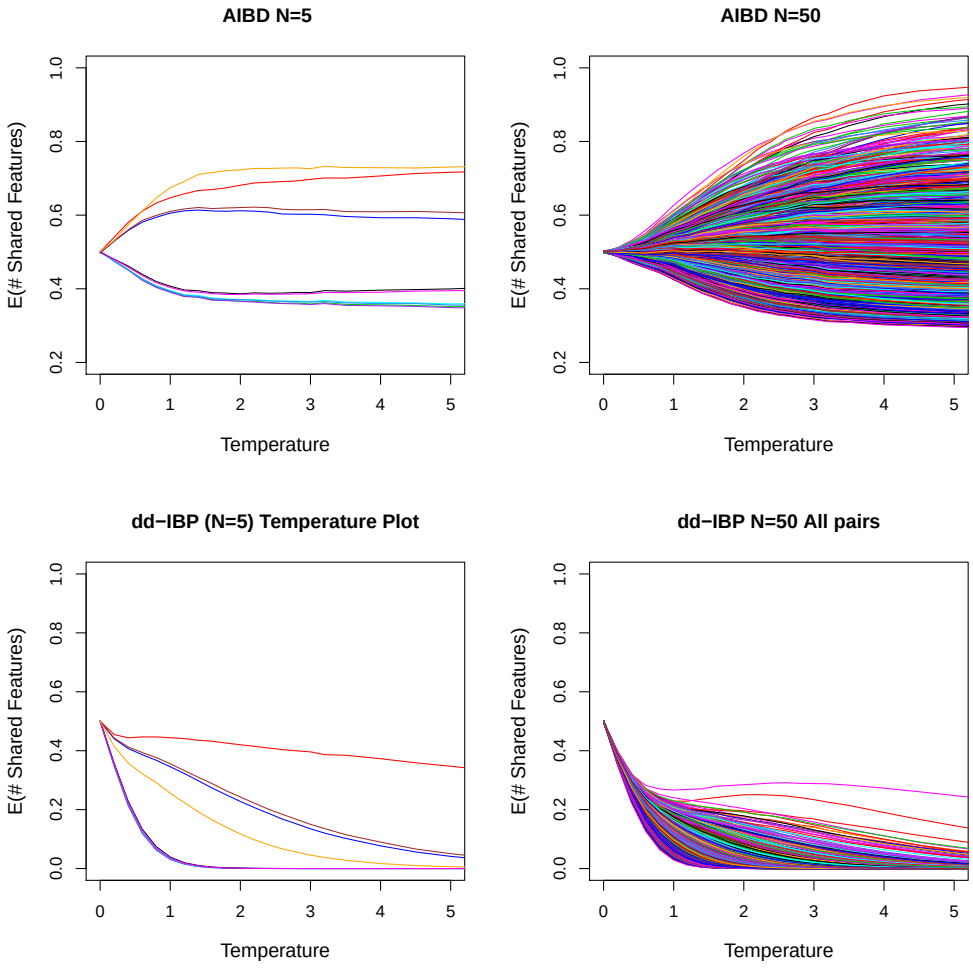


Figure 3: A view of the average feature sharing as a function of temperature for the AIBD and dd-IBP using sample sizes of $N = 5$ and $N = 50$, where $\alpha = 1$. This figure shows how the AIBD retains the overall average of the expected number of shared features of the IBP, while on average the dd-IBP with a sequential proximity matrix shares less, as the temperature increases. For both sample sizes and the AIBD and the dd-IBP the permutation of the data, $\boldsymbol{\rho}$, was integrated out of the results similar to Figure 2 but using Monte Carlo integration.

Property	AIBD	dd-IBP
Explicit pmf	Yes	No
Reduces to IBP when $\tau = 0$	Yes	In one case
E(# Features)	Same as IBP	Different from IBP
E(# Features per customer)	Same as IBP	Different from IBP
E(# of Total shared features)	Same as IBP*	Different from IBP**
E(# Shared features per customer)	Higher or Lower	Different from IBP**

Table 2: Summary of the properties of the AIBD and dd-IBP to the IBP. Details of the first two properties can be found in Section 3, and the remainder can be found in Section 4, with a focus on Section 4.3 for the dd-IBP's properties.
$*$ Demonstrated via simulation.
$**$ We consistently found this to be lower than the IBP in simulations.

4.3 Comparison of the AIBD's and the dd-IBP's Properties

Although both use the pairwise distances of items, the AIBD and the dd-IBP have some notable differences. To compare the AIBD's properties to the dd-IBP's, we refer to Figure 3. The bottom row of the figure shows that as the temperature increases in the dd-IBP, on average, pairs of customers share less. Asymptotically, all average feature sharing goes to zero; that is, at a temperature of infinity, all customer pairs will share no features.

A table comparing several properties of the AIBD and dd-IBP to the IBP is shown in Table 2. In the dd-IBP, on average the customers share less as τ increases. As a result, all lines in the dd-IBP plots in Figure 3 go to zero as $\tau \to \infty$. This may be sensible in cases where we want customers to share less than the IBP. However, it does not seem possible to allow certain customer pairs to share more than the IBP. In contrast, the AIBD upweights or downweights average sharing depending on the pairwise similarity value while still having the same average expected number of shared features across all pairs as the IBP. Larger N increases the disparity between pairs in the AIBD, but appears to decrease the disparity between pairs for the dd-IBP.

For the dd-IBP, a major consequence of the expected number of shared features going to zero as $\tau \to \infty$ is that the total number of features in the distribution increases to αN. For $N = 50$ and $\alpha = 1$, the expected number of features in the AIBD is the 50th harmonic number, or 4.5 for all temperatures. For the dd-IBP in this example, however, the expected number of features can range from 4.5 to 50, depending on the temperature chosen. This is shown in Figure 4. The dd-IBP reduces sharing and borrowing of strength between features, which is a primary reason to use the IBP. This can potentially result in higher computational burdens when a $\boldsymbol{Z}$ from the dd-IBP has a larger number of columns than the AIBD. One reason why the average number of shared features in the AIBD does not tend to zero is that the AIBD preserves the distribution of the total number of features for fixed α and N. Thus, in the AIBD, the distribution of the number of columns is unchanged from the IBP. The AIBD only influences how features are shared.

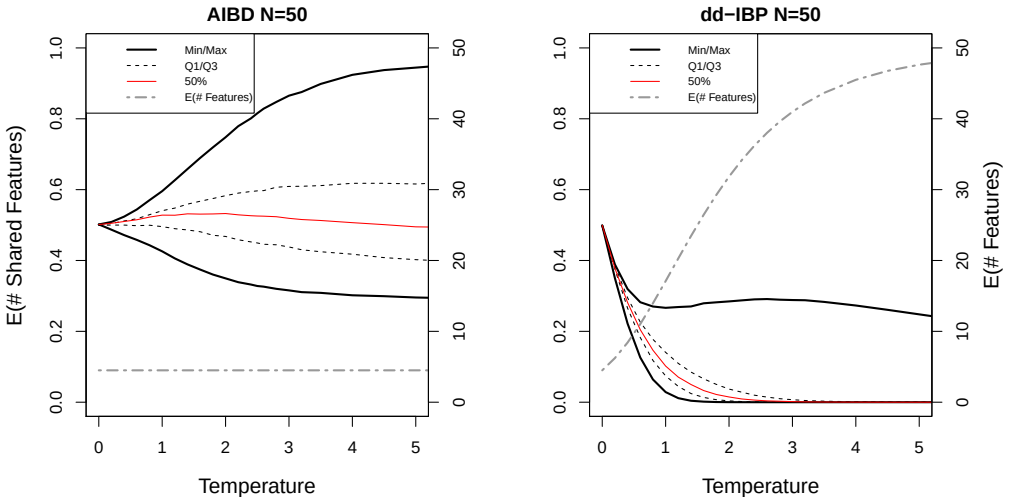


Figure 4: The distribution of the total number features overlayed with the temperature plots for the AIBD and the dd-IBP for $N = 50$. The Q1 and Q3 lines represent the quartiles of the number of shared features. This figure highlights that as the temperature increases, the average number of features (gray dashed line) remain fixed in the AIBD, but grows rapidly in the dd-IBP. It also shows how the median number of shared features (red line) of the AIBD and the dd-IBP change as a function of temperature.

5 AIBD Posterior Sampling

We now describe how to sample from the joint posterior distribution of the model parameters in the LGLFM (i.e., $\boldsymbol{Z}$, τ, α, $\boldsymbol{\rho}$, σ_A and σ_X), presented in Section 2.3, with an AIBD prior on the feature allocation. . We suggest priors for the parameters and a Metropolis-Hastings-within-Gibbs sampling algorithm. Using the likelihood in Equation (4), we can write the full joint posterior as shown in the following equation:

$$p(\boldsymbol{Z}, \boldsymbol{\rho}, \alpha, \tau, \sigma_X, \sigma_A | \boldsymbol{X}, \boldsymbol{\Lambda}) \propto p(\boldsymbol{Z}|\alpha, \boldsymbol{\rho}, \tau)\, p(\boldsymbol{\rho})\, p(\alpha)\, p(\tau)\, p(\sigma_X, \sigma_A)\, p(\boldsymbol{X}|\boldsymbol{Z}, \sigma_X, \sigma_A). \tag{9}$$

5.1 Posterior Sampling of the Feature Allocation Z

We use an AIBD prior on $\boldsymbol{Z}$ and suggest the following algorithm to sample from the posterior. From Equation (9), the full conditional distribution of $\boldsymbol{Z}$ is proportional to $p(\boldsymbol{Z}|\alpha, \boldsymbol{\rho}, \tau)\, p(\boldsymbol{X}|\boldsymbol{Z}, \sigma_X, \sigma_A)$. Instead of proposing an entirely new $\boldsymbol{Z}$ matrix, we update $\boldsymbol{Z}$ row-by-row. Define a singleton feature for row i to be a feature that only customer i has. In other words, a singleton feature for customer i is a column in $\boldsymbol{Z}$ of all zeros except for a 1 on row i. Define the non-singleton features to be features that are owned by any of the other customers. For each row $i \in \{1, 2, ..., N\}$ in $\boldsymbol{Z}$:

1. Let $m_1, m_2, ...m_h \in \mathcal{K}$ be the collection of column indices of the non-singleton features in the current state of $\boldsymbol{Z}$, where h is the total number of non-singletons. If $h = 0$, skip to step 3. If $h > 0$, generate a random permutation of the collection of indices in $\mathcal{K}$. We update the non-singleton columns in this order in step 2.

2. Start by updating the non-singleton features on row i one at a time using the Metropolis-Hastings algorithm in the permuted order as generated in step 1. Denote $z_{i,m}$ to be the binary number in the i^{th} row and m^{th} column of $\boldsymbol{Z}$. We update each element $z_{i,m}$ for each $m \in \mathcal{K}$ sequentially according to the permutation of $\mathcal{K}$. For each $m \in \mathcal{K}$:

   (a) Define the active feature to be the feature in the m^{th} column of the current state of $\boldsymbol{Z}$. This is the feature that is currently being updated.

   (b) Propose $z^*_{i,m} = 1 - z_{i,m}$ (i.e., the opposite of the current state of $z_{i,m}$). Let $\boldsymbol{Z}^*$ be the same as $\boldsymbol{Z}$, except for $z^*_{i,m}$ in place of $z_{i,m}$.

   (c) Since the order of the columns does not matter, different updates may result in the same proposed feature allocation. Thus, the proposal distribution is not symmetric. The Hastings ratio is computed by dividing the number of features identical to the active feature d^* in $\boldsymbol{Z}^*$ by the number of features identical to the active feature d in $\boldsymbol{Z}$. The active feature is also counted in this total, so if the active feature is distinct in both the current and proposed states, the ratio is 1. From the example in Figure 5, $m = 4$ and $h = 4$, and the active feature is column 4. In the current state, there are $d = 2$ features identical to column 4 (including itself), and $d^* = 3$ features identical to column 4 in the proposed state.

Current State $\boldsymbol{Z}$

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

Proposed State $\boldsymbol{Z}^*$

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

Figure 5: Example of a posterior update from step 2. Customer 1 has one singleton feature (column 3) and four non-singleton features. For customer 1, we will individually update the columns 1,2,4, and 5 in a random order. The underlined number z_{14} is being updated and thus the active feature is column 4. There are 3 features identical to the active feature (including itself) in the proposed state and 2 identical features in the current state. The Hastings ratio is 3/2 to account for the asymmetric proposal.

(d) Compute the Metropolis-Hastings ratio $MHR = \frac{p(\boldsymbol{Z}^*|\alpha,\boldsymbol{\rho},\tau)p(\boldsymbol{X}|\boldsymbol{Z}^*,\sigma_X,\sigma_A)}{p(\boldsymbol{Z}|\alpha,\boldsymbol{\rho},\tau)p(\boldsymbol{X}|\boldsymbol{Z},\sigma_X,\sigma_A)} \frac{d^*}{d}$ and update $z_{i,m}$ to $z^*_{i,m}$ with probability min(1, MHR), else leave $z_{i,m}$ unchanged.

3. Now we propose new singleton features for customer i.

   (a) We first evaluate the unnormalized probability mass of the full conditional distribution of $\boldsymbol{Z}$ after adding $0, 1, 2, \ldots$ features (with all other parameters and rows in $\boldsymbol{Z}$ held constant). Since we cannot check a theoretically infinite number of features, we stop considering new features once we obtain a mass that is less than a specific fraction (e.g., we used 1/1000) of the highest mass. This should cover most reasonable posterior values and the fraction is a tuning parameter that can be adjusted if desired. This truncation step makes this algorithm an approximate sampler, but it should closely mimic an exact sampler.

   (b) We estimate probabilities $p_0, p_1, p_2, ...$ of adding 0,1,2,... singleton features by dividing each unnormalized mass in the previous step by the sum of all masses. Add j singletons to customer i with probability p_j.

After going through all rows, the result is one scan of the Markov chain updates for $\boldsymbol{Z}$. We then sample from the other parameters, which we describe in the next section.

5.2 Sampling the Other Parameters

After updating $\boldsymbol{Z}$, we proceed to update the other parameters (α, $\boldsymbol{\rho}$, τ, σ_X, and σ_A). The parameters α and τ are sampled univariately; while (σ_X, σ_A) is sampled jointly, and $\boldsymbol{\rho}$ sampled as a vector.

For the mass parameter α, we suggest using a gamma prior because it is conditionally conjugate. If a Gamma(a_α,b_α) prior (with expectation a_α/b_α) is used, then the resulting conditional posterior is a draw from a Gamma($a_\alpha + T_{\boldsymbol{Z}}$, $b_\alpha + H_N$). $T_{\boldsymbol{Z}}$ indicates the total number of features in the current state of $\boldsymbol{Z}$ and H_N is the N^{th} harmonic number.

For the permutation parameter $\boldsymbol{\rho}$, we suggest using a discrete uniform on all possible permutations of the integers 1 through N, unless there is natural ordering in the data. Using the discrete uniform prior, we update $\boldsymbol{\rho}$ with a random walk using a discrete uniform proposal. For small N, this can be done by sampling from any of the $N!$ possible permutations. However, for larger N, this could lead to low acceptance rates. As such, we recommend only updating k_ρ of the elements in the permutation at a time, where k_ρ is a tuning parameter that controls how quickly the permutation space is explored. The steps to sample a new $\boldsymbol{\rho}$ are outlined as follows. First randomly select k_ρ indices in the permutation to update. Next, randomly shuffle the k_ρ indices, while leaving the other $N - k_\rho$ indices fixed, to generate a proposed permutation. Finally, calculate the Metropolis acceptance ratio, R_ρ and accept the proposed permutation with probability $\min(1, R_\rho)$.

The temperature parameter τ in the likelihood does not appear to have a conjugate prior. Therefore, any prior with positive continuous support might be reasonable. For our implementation, we choose a Gamma prior. To draw from the conditional posterior of τ we use a Metropolis step with a Gaussian random walk proposal and reject proposals outside the support of τ.

For the variance components of the likelihood, we recommend using any positive continuous prior, such as a Gamma prior on σ_X and σ_A. Since σ_X and σ_A are typically negatively correlated in the posterior, we used a bivariate Gaussian random walk to update both parameters simultaneously, again rejecting proposals outside the support of (σ_A, σ_X).

Due to the computational burden of updating $\boldsymbol{Z}$ relative to updating the other parameters, we recommend updating the other parameters several times for every update of $\boldsymbol{Z}$. We updated other parameters 10 times for every update of $\boldsymbol{Z}$ in the application in Section 6 to reduce the autocorrelation within the posterior draws, at a negligible computational cost.

The methods suggested in this section are implemented in the *samplePosteriorLGLFM* function of the *aibd* R package. The posterior sampling algorithm can be applied to both the AIBD and, since it is a special case of the AIBD, the IBP. From our experience, the results are accurate and the only source of bias comes from the truncation step. This truncation error can be controlled and is negligible compared to Monte Carlo error.

6 Data Analysis

In this section, brain imaging data is analyzed using the LGLFM (Griffiths and Ghahramani, 2006) discussed in Section 2.3. We use the AIBD, dd-IBP, and standard IBP as priors for the feature allocation matrix $\boldsymbol{Z}$ in the LGLFM, and compare the posterior inferences under the different priors

6.1 The Data

The data for these analyses were obtained from a neuroimaging study of the brains of healthy and Alzheimer's-diseased subjects (see Dinov et al., 2009). The data and some details of the study are available on UCLA's Statistics Online Computational Resource (SOCR) data page SOCR (2009). Of that data, we consider the 27 Alzheimer's diseased and 35 normal control subjects. In the study, 56 distinct regions of interest (ROIs) in the brain are observed; each region has four different measurements. The four measurements are the surface area (SA), shape index (SI), curvedness (CV), and fractal dimension (FD). One of the properties of the LGLFM is that the error terms for each of these measurements are assumed to have constant variance. Therefore, before modeling, the ROI measurements are centered and scaled (so each has a mean of zero and a standard deviation of one). Additionally, the SA measurements are somewhat skewed, thus a log transformation is performed before centering and scaling those measurements.

6.2 The Analysis

The analysis of this data mirrors the steps taken in Gershman et al. (2015). First, patient age is included in the AIBD and dd-IBP priors as a distance between patients. Including this distance makes the both the AIBD and dd-IBP priors nonexchangeable, with the hope that this extra information will improve the model's predictive performance. The distances between patients are defined in both priors using the exponential decay function (i.e., $\exp\{-\text{temperature} \times |\text{age difference}|\}$)[1]. As in their analysis, we use a sequential proximity matrix for the dd-IBP and assume the sequence is the ordering of the patients in the data. The temperatures for the AIBD and dd-IBP are set at 5 levels (0.4, 0.8, 1.2, 1.6, and 2.0). Since the IBP prior is exchangeable, it does not include any distance information between patients. The mass parameter is set to 10 for each prior (which is the default value in the dd-IBP code by Gershman, 2013). The LGLFM likelihood is used as defined previously, with the data being the 4 different measurements in 56 regions of the brain. The data, $\boldsymbol{X}$, are contained in a 62×224 matrix, which is an over parameterized model unless some type of regularization or dimension reduction is used.

The prior distributions for the variance components of the likelihood also need to be specified. Since the data are centered and scaled, a reasonable maximum value for σ_A and σ_X is 1. Therefore, in the models with AIBD or IBP priors on $\boldsymbol{Z}$, a standard uniform prior is placed on these parameters; i.e., $p(\sigma_A^2) = \mathrm{I}\,(0 < \sigma_A^2 < 1)$ and $p(\sigma_A^2) = \mathrm{I}\,(0 < \sigma_X^2 < 1)$. The available code which implements posterior inference for the dd-IBP places a similar flat prior on these parameters; namely $p(\sigma_A^2) \propto \mathrm{I}\,(0 < \sigma_A^2)$ and $p(\sigma_X^2) \propto \mathrm{I}\,(0 < \sigma_X^2)$.

With the models fully specified, we follow the steps of the analysis in Gershman et al. (2015). First, we obtain 1,500 MCMC posterior samples from each model. Among those posterior samples the *maximum a posteriori* (MAP) estimate is selected. Then the subjects are randomly divided into training and testing sets. Using the latent features from the

[1] In the AIBD prior, 0.00001 is added to each age difference to ensure no two patients have a distance of 0.

MAP estimate as predictors, an L_2-regularized logistic regression is performed on the training set (to classify which patients do or do not have Alzheimer's disease). For the logistic regression, we use the *penalized* function in the *penalized R* package (Goeman et al., 2018; Goeman, 2010). Using the results from the logistic regression model, the test group subjects are then classified to assess performance. Finally, for each test group's classification, the area under the receiver operating characteristic curve (AUC) is calculated. The AUC is the metric used to compare each model's efficacy.

Although the steps in this analysis are the same as in Gershman et al. (2015), based on the information provided in that article, several aspects of the analysis cannot be replicated. First, the mass parameter is not specified in Gershman et al. (2015). Thus, we assume it is fixed at 10 as in the default of the available dd-IBP code. Next, in the dd-IBP paper a few observations are randomly removed in the classification to balance the training and testing sets. However, the removed observations are not identified. Therefore, we do not ignore any observations; i.e., the training and testing sets are disjoint but include all 62 subjects. Also, it is not specified which observations are assigned to the training and testing sets for classification. In our implementation, we randomly assign 14 diseased and 18 healthy subjects to the training set, and then assign the remaining 13 diseased and 17 healthy subjects to the testing set; this is repeated for each of the 50 MCMC runs (e.g., each models' MAP estimate from their i^{th} MCMC run uses the same training and testing sets for classification).

In total there are five AIBD priors (one for each temperature), five dd-IBP priors, and one IBP prior. Each of those 11 models has 50 independent MCMC runs; the i^{th} runs of the three models are compared using identical training and testing sets during classification. For posterior simulation, the code available at Gershman (2013) is used for the dd-IBP, and the functions included in the *aibd* package (Dahl et al., 2020) are used for both the AIBD and the IBP. One possible confounding factor in our comparison of the AIBD and the dd-IBP is that the posterior simulation for the dd-IBP is an approximate MCMC scheme, as noted in Gershman (2013).

6.3 Comparison Between the AIBD and the dd-IBP

Each model's performance is measured by how well it correctly classifies subjects into "healthy" or "diseased". This performance can be quantified using the AUC, which ranges between zero and one with higher values indicating a better classifier. Using the AUC, from the classifications on the testing sets, the results of the models with AIBD and dd-IBP priors are compared. The AUC (averaged over the 50 runs) for the models with AIBD and dd-IBP priors are reported in Table 3.

Table 3 shows that, on average, the AIBD has higher AUC than the dd-IBP at every temperature setting. To make a more formal comparison, we compute confidence intervals on the paired differences. Since the i^{th} MCMC run, for each model, is assigned the same training and testing sets the pairing is done on the MCMC run number. The confidence intervals for the average differences in AUC (AIBD - dd-IBP) are reported in Table 4.

Average AUC

	AIBD prior	dd-IBP prior
Temp = 0.4	0.7470	0.6800
Temp = 0.8	0.7510	0.7111
Temp = 1.2	0.7489	0.6745
Temp = 1.6	0.7552	0.7437
Temp = 2.0	0.7542	0.7051

Table 3: The average AUC from the resulting classifications of the 10 different models, higher values indicate better model performance.

	Mean Estimated Difference	Lower Bound	Upper Bound
Temp = 0.4	0.0670	0.0352	0.0987
Temp = 0.8	0.0399	0.0003	0.0795
Temp = 1.2	0.0743	0.0361	0.1125
Temp = 1.6	0.0114	-0.0306	0.0535
Temp = 2.0	0.0490	0.0195	0.0786

Table 4: 95% confidence intervals for the average difference of AUC from the AIBD and dd-IBP at five different temperatures, higher values indicate better model performance.

As shown in Table 4, on average the AIBD significantly (at the 95% level) outperforms the dd-IBP (with respect to AUC) in 4 of the 5 temperature settings. The temperature of 1.6 shows a slight improvement using the AIBD versus the dd-IBP, but statistical significance is not reached.

While the AIBD performs better than the dd-IBP under these conditions, it has the added benefit of easily accommodating a prior on both the temperature and the mass parameter. Finding suitable static values for either of these parameters seems neither intuitive nor straight-forward. Allowing prior distributions to incorporate some amount of uncertainty is arguably more appropriate. An additional model with an AIBD prior was fit (with 50 independent MCMC chains) which has a prior distribution set on the mass and temperature parameters. When taking advantage of this flexibility the AIBD performs even better with an average AUC of 0.7737 (over 50 runs). A Gamma(1,1) prior distribution is used for both the mass and temperature parameters in the AIBD model. Confidence intervals on the average difference in AUC for the AIBD model (with priors set on the mass and temperature parameters) and the 5 dd-IBP models are compared in Table 5.

In this example, Table 5 provides an even stronger argument that the AIBD prior is better able to capture the distance information (age) between subjects. On average, the logistic regression trained on the $\boldsymbol{Z}$ learned from the AIBD model, when compared to the dd-IBP model, more accurately classifies subjects into "healthy" and "diseased." We also ran an additional simulation for the dd-IBP using a symmetric proximity matrix to determine if the ordering of the patients negatively impacted the dd-IBP's results. The results were similar to using a sequential proximity matrix, with a slight improvement of roughly 0.008 increase in average AUC.

	Mean Estimated Difference	Lower Bound	Upper Bound
Temp = 0.4	0.0936	0.0567	0.1305
Temp = 0.8	0.0626	0.0207	0.1045
Temp = 1.2	0.0991	0.0565	0.1417
Temp = 1.6	0.0300	-0.0086	0.0685
Temp = 2.0	0.0686	0.0350	0.1021

Table 5: 95% confidence intervals for the difference in AUC of the AIBD model (random temp and mass) and the five dd-IBP models (fixed temps and mass), higher values indicate better model performance.

Over the five selected temperatures and 50 MCMC chains, the average number of posterior MAP features in $\boldsymbol{Z}$ was 64.4 for the dd-IBP and 65.1 for the AIBD. These numbers demonstrate a substantial reduction in dimension from the original 224 data columns plus the age variable. This reduction in dimension is really larger than it first appears because the entries in feature allocation matrices are binary, zeros and ones, whereas the data are continuous values. To determine how much information from the data is lost in the dimension reduction, we compare the AUC results from two models. The first model is a penalized logistic regression built off the MAP feature allocation from the LGLFM using an AIBD prior (with random mass and temperature). The second model is a typical penalized logistic regression model fit to a training set of the centered and scaled data and the age variable. We compared these two models 50 times and paired the results with models that used common training and testing sets. The AUC associated with the AIBD's classifiers nearly matches the AUC of the classifiers from the original data. A 95% confidence interval of the difference in AUCs is (-0.042, 0.003) with a mean estimate of -0.020. So little information is lost in the dimension reduction and the AIBD's MAP feature allocation can comprise nearly all the information about a patient's disease state, while drastically decreasing the dimensionality. This reduction of dimension is most useful if the results can be interpreted.

A very rough speed comparison of the posterior sampling algorithms for the dd-IBP and the AIBD showed that the dd-IBP could do one update of all parameters in roughly 18 seconds in CPU time, while the AIBD took approximately 70 seconds. There are a few issues that make a full comparison quite challenging. One is that the methods are implemented in different software; the dd-IBP takes advantage of MATLAB's multi-threading capabilities, while the AIBD implementation is single-threaded. Additionally, the dd-IBP sampling algorithm is a heuristic while the AIBD implementation is a valid MCMC scheme.

6.4 Comparison Between the AIBD and the IBP

As demonstrated, the AIBD competes favorably with the dd-IBP as an effective nonexchangeable prior for latent feature models. In this application we also want to show that using a nonexchangable prior can produce better results than when using an exchangeable prior, specifically the IBP. In this case the AIBD has the added advantage over the IBP of knowing a patient's age and using it as a distance between patients. If a patient's

age is important to predicting Alzheimer's disease the AIBD should outperform the IBP; however, if the information is not informative no benefit should be expected.

The comparison between the models with the AIBD and IBP priors uses the same scheme as in the previous subsections. The prior for the mass parameter in both models is a Gamma(1,1) distribution. Additionally, a Gamma(1,1) prior is set on the temperature parameter in the AIBD model. The resulting average AUC for each model is 0.7737 and 0.7301 for the AIBD and IBP, respectively. A 95% confidence interval on the paired difference between average AUC for the AIBD and the IBP is $(0.0203, 0.0668)$. The results are fairly convincing: the AIBD can use the extra distance information to obtain better classification results.

This example demonstrates that a patient's age does contain information that is valuable in classifying patients into healthy and Alzheimer's diseased states. It also shows that the AIBD prior is able to capture this distance information and improve model performance.

A rough speed comparison of the posterior sampling algorithms for the IBP and the AIBD showed that the IBP could do one update of all parameters in roughly 63 seconds in CPU time, while the AIBD took approximately 70 seconds. This comparison is easier to make than with the dd-IBP, since the algorithm and software implementation are the same. This application demonstrates that the computational penalty for using the nonexchangeable AIBD prior isn't high, and improves model performance over the IBP prior.

7 Conclusion

In this paper, a generalization of the IBP was developed to allow for customer dependence in the prior. It was demonstrated that after including pairwise distance information, the AIBD preserves many familiar properties of the IBP. We compared these properties of the AIBD to those of the IBP and dd-IBP, and summarized them in Table 2. Futher, while the AIBD and dd-IBP attempt to generalize the IBP by including pairwise distance information, we have shown that the AIBD possesses several properties that make it particularly appealing. An instance of this was shown in the application in Section 6, where the AIBD outperformed the dd-IBP in terms of AUC.

Overall, the AIBD is an attractive solution for incorporating distance information into a prior distribution of a feature allocation. It retains many desirable properties of the IBP. For a fixed mass and temperature, it encourages more feature sharing between customers than the dd-IBP. Priors can readily be set on model parameters, such as the mass and temperature. Last, but not least, this distribution and some associated methods are implemented in the *aibd* package in R (Dahl et al., 2020).

References

Blei, D. M. and Frazier, P. I. (2011). "Distance dependent Chinese restaurant processes." *Journal of Machine Learning Research*, 12(Aug): 2461–2488. 3

Chen, M., Gao, C., and Zhao, H. (2013). "Phylogenetic Indian buffet process: Theory and applications in integrative analysis of cancer genomics." *arXiv preprint arXiv:1307.8229*. 5

Dahl, D. B., Day, R., and Tsai, J. W. (2017). "Random Partition Distribution Indexed by Pairwise Information." *Journal of the American Statistical Association*, 112(518): 721–732.
URL https://doi.org/10.1080/01621459.2016.1165103 3

Dahl, D. B., Warr, R., and Meyer, J. (2020). *aibd: Attraction Indian Buffet Distribution*. R package version 0.1.8.
URL https://CRAN.R-project.org/package=aibd 20, 23

Dinov, I., Van Horn, J., Lozev, K., Magsipoc, R., Petrosyan, P., Liu, Z., MacKenzie-Graha, A., Eggert, P., Parker, D. S., and Toga, A. W. (2009). "Efficient, distributed and interactive neuroimaging data analysis using the LONI pipeline." *Frontiers in neuroinformatics*, 3: 22. 2, 19

Gershman, S. J. (2013). "Matlab code for the Distance Dependent Infinite Latent Feature Models article." http://gershmanlab.webfactional.com/pubs/ddIBP_release.zip. Accessed: 2017-11-15. 19, 20

Gershman, S. J., Frazier, P. I., and Blei, D. M. (2015). "Distance dependent infinite latent feature models." *IEEE transactions on pattern analysis and machine intelligence*, 37(2): 334–345. 2, 5, 7, 9, 19, 20

Goeman, J. J. (2010). "L1 penalized estimation in the Cox proportional hazards model." *Biometrical Journal*, 1(52): 70–84. 20

Goeman, J. J., Meijer, R. J., and Chaturvedi, N. (2018). *Penalized: L1 (lasso and fused lasso) and L2 (ridge) penalized estimation in GLMs and in the Cox model*. R package version 0.9-51. 20

Griffiths, T. L. and Ghahramani, Z. (2005). "Infinite latent feature models and the Indian buffet process." Technical Report 2005-001, Gatsby Computational Neuroscience Unit. 6

— (2006). "Infinite latent feature models and the Indian buffet process." In *Advances in neural information processing systems*, 18, 475–482. Cambridge, MA: MIT Press. 3, 18

— (2011). "The Indian buffet process: An introduction and review." *Journal of Machine Learning Research*, 12(Apr): 1185–1224. 1, 6

Hai-son, P. L. and Bar-Joseph, Z. (2011). "Inferring interaction networks using the ibp applied to microrna target prediction." In *Advances in Neural Information Processing Systems*, 235–243. 5

Lee, J., Müller, P., Gulukota, K., Ji, Y., et al. (2015). "A Bayesian feature allocation model for tumor heterogeneity." *The Annals of Applied Statistics*, 9(2): 621–639. 5

Lee, J., Müller, P., Sengupta, S., Gulukota, K., and Ji, Y. (2016). "Bayesian inference for intratumour heterogeneity in mutations and copy number variation." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(4): 547–563. 5

Lui, A., Lee, J., Thall, P. F., Daher, M., Rezvani, K., and Barar, R. (2020). "A Bayesian Feature Allocation Model for Identification of Cell Subpopulations Using Cytometry Data." *arXiv preprint arXiv:2002.08609*. 5

Miller, K. T., Griffiths, T., and Jordan, M. I. (2012). "The Phylogenetic Indian Buffet Process: A Non-Exchangeable Nonparametric Prior for Latent Features." 5

Ni, Y., Mueller, P., and Ji, Y. (2018). "Bayesian Double Feature Allocation for Phenotyping with Electronic Health Records." *arXiv preprint arXiv:1809.08988*. 5

Sengupta, S., Wang, J., Lee, J., Müller, P., Gulukota, K., Banerjee, A., and Ji, Y. (2014). "Bayclone: Bayesian nonparametric inference of tumor subclones using ngs data." In *Pacific Symposium on Biocomputing Co-Chairs*, 467–478. World Scientific. 5

SOCR (2009). "SOCR Data July2009 ID NI." http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_July2009_ID_NI. Accessed: 2019-04-15. 19

Tan, X., Rao, V., and Neville, J. (2018). "The Indian Buffet Hawkes Process to Model Evolving Latent Influences." In *UAI*, 795–804. 6

Williamson, S., Orbanz, P., and Ghahramani, Z. (2010). "Dependent Indian Buffet Processes." In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, 924–931. Chia Laguna Resort, Sardinia, Italy: PMLR. URL http://proceedings.mlr.press/v9/williamson10a.html 5

Williamson, S. A., Zhang, M. M., and Damien, P. (2020). "A New Class of Time Dependent Latent Factor Models with Applications." *Journal of Machine Learning Research*, 21(27): 1–24. 6

Xu, Y., Lee, J., Yuan, Y., Mitra, R., Liang, S., Müller, P., and Ji, Y. (2013). "Nonparametric bayesian bi-clustering for next generation sequencing count data." *Bayesian analysis (Online)*, 8(4): 759. 5

Xu, Y., Müller, P., Yuan, Y., Gulukota, K., and Ji, Y. (2015). "MAD Bayes for tumor heterogeneity—feature allocation with exponential family sampling." *Journal of the American Statistical Association*, 110(510): 503–514. 5

Acknowledgments

This work was supported, in part, by NIH NIGMS R01 GM104972.