

Distance Dependent Infinite Latent Feature Models

Samuel J. Gershman¹, Peter I. Frazier² and David M. Blei³

¹ Department of Psychology and Princeton Neuroscience Institute,
Princeton University

² School of Operations Research and Information Engineering,
Cornell University

³ Department of Computer Science,
Princeton University

Abstract

Latent feature models are widely used to decompose data into a small number of components. Bayesian nonparametric variants of these models, which use the Indian buffet process (IBP) as a prior over latent features, allow the number of features to be determined from the data. We present a generalization of the IBP, the *distance dependent Indian buffet process* (dd-IBP), for modeling non-exchangeable data. It relies on distances defined between data points, biasing nearby data to share more features. The choice of distance measure allows for many kinds of dependencies, including temporal and spatial. Further, the original IBP is a special case of the dd-IBP. In this paper, we develop the dd-IBP and theoretically characterize its feature-sharing properties. We derive a Markov chain Monte Carlo sampler for a linear Gaussian model with a dd-IBP prior and study its performance on several non-exchangeable data sets.

KEYWORDS: Bayesian nonparametrics, dimensionality reduction, matrix factorization

1 Introduction

Many natural phenomena decompose into latent features. For example, visual scenes can be decomposed into objects; genetic regulatory networks can be decomposed into transcription factors; music can be decomposed into spectral components. In these examples, multiple latent features can be simultaneously active, and each can influence the observed data. Dimensionality reduction methods, such as principal component analysis, factor analysis, and probabilistic matrix factorization, provide a statistical approach to inferring latent features [4]. These methods characterize a small set of dimensions, or features, and model each data point as a weighted combination of these features. Dimensionality reduction can improve predictions and identify hidden structures in observed data.

Dimensionality reduction methods typically require that the number of latent features (i.e., the number of dimensions) be fixed in advance. Researchers have recently proposed a more flexible approach based on Bayesian nonparametric models, where the number of features is inferred from the data through a posterior distribution. These models are usually based on the Indian buffet process (IBP; [16, 17]), a prior over binary matrices with a finite number of rows (corresponding to data points) and an infinite number of columns (corresponding to latent features). Using the IBP as a building block, Bayesian nonparametric latent feature models have been applied to several statistical problems (e.g., [19, 20, 22, 23]). Since the number of features is effectively unbounded, these models are sometimes known as “infinite” latent feature models.

The IBP assumes that data are *exchangeable*: permuting the order of rows leaves the probability of a feature matrix unchanged. This assumption may be appropriate for some data sets, but for many others we expect dependencies between data points and, consequently, between their latent representations. As examples, the latent features describing human motion are autocorrelated over time; the latent features describing environmental risk factors are autocorrelated over space. In this paper, we present a generalization of the IBP—the *distance dependent IBP* (dd-IBP)—that addresses this limitation. The dd-IBP allows infinite latent feature models to capture non-exchangeable structure.

The problem of adapting nonparametric models to non-exchangeable data has been studied extensively in the mixture-modeling literature. In particular, variants of the Dirichlet process mixture model allow dependencies between data points (e.g., [9, 15, 7, 12, 24, 1]). These dependencies may be spatial, temporal or more generally covariate-dependent; the effect of such dependencies is to induce sharing of features between nearby data points.

Among these methods is the *distance dependent Chinese restaurant process* (dd-CRP; [6]). The dd-CRP is a non-exchangeable generalization of the Chinese restaurant process (CRP), the prior over partitions of data that emerges in Bayesian nonparametric mixture modeling [5, 14, 25]. The dd-CRP models non-exchangeability by using distances between data points—nearby data points (e.g., in time or space) are more likely to be assigned to the same mixture component. The dd-IBP extends these ideas to infinite latent feature models, where distances between data points influence feature-sharing, and nearby data points are more likely to share latent features.

We review the IBP in Section 2.1 and develop the dd-IBP in Section 2.2. Like the dd-CRP, the dd-IBP lacks *marginal invariance*, which means that removing one observation changes the

distribution over the other observations. We discuss this property further in Section 2.3. Although many Bayesian nonparametric models have this property, we view it as a particular modeling choice that may be appropriate for some problems but not for others.

Several other infinite latent feature models have been developed to capture dependencies between data in different ways, for example using phylogenetic trees [21] or latent Gaussian processes [29]. Of particular relevance to this work is the model of Zhou et al. [31], which uses a hierarchical beta process to couple data. These and other related models are discussed further in Section 3. In Section 4, we characterize the feature-sharing properties of the dd-IBP and compare it to those of the model proposed by Zhou et al. [31]. We find that the different models capture qualitatively distinct dependency structures.

Exact posterior inference in the dd-IBP is intractable. We present an approximate inference algorithm based on Markov chain Monte Carlo (MCMC; [27]) in Section 5, and we apply this algorithm in Section 6 to infer the latent features in a linear-Gaussian model. The experimental results presented in Section 7 suggest that the dd-IBP is an effective tool for modeling latent structure in data with dependencies between observations.

2 The distance dependent Indian buffet process

We first review the definition of the IBP and its role in defining infinite latent feature models. We then introduce the dd-IBP.

2.1 The Indian buffet process

The IBP is a prior over binary matrices \mathbf{Z} with an infinite number of columns [16, 17]. In the Indian buffet metaphor, rows of \mathbf{Z} correspond to customers and columns correspond to dishes. In data analysis, the customers represent data points and the dishes represent features. Let z_{ik} denote the entry of \mathbf{Z} at row i and column k . Whether customer i has decided to sample dish k (that is, whether $z_{ik} = 1$) corresponds to whether data point i possesses feature k .

The IBP is defined as a sequential process. The first customer enters the restaurant and samples the first $\lambda_1 \sim \text{Poisson}(\alpha)$ number of dishes, where the hyperparameter α is a scalar. In the binary matrix, this corresponds to the first row being a contiguous block of ones, whose length is the number of dishes sampled (λ_1), followed by an infinite block of zeros.

Subsequent customers $i = 2, \dots, N$ enter, sampling each previously sampled dish according to its popularity,

$$p(z_{ik} = 1 \mid \mathbf{z}_{1:(i-1)}) = m_{ik}/i, \tag{1}$$

where $m_{ik} = \sum_{j < i} z_{jk}$ is the number of customers that sampled dish k prior to customer i . (We emphasize that Eq. 1 applies only to dishes k that were previously sampled, i.e., for which $m_{ik} > 0$.) Then, each customer samples $\lambda_i \sim \text{Poisson}(\alpha/i)$ new dishes. Again these are represented as a contiguous block of ones in the columns beyond the last dish sampled by a previous customer.

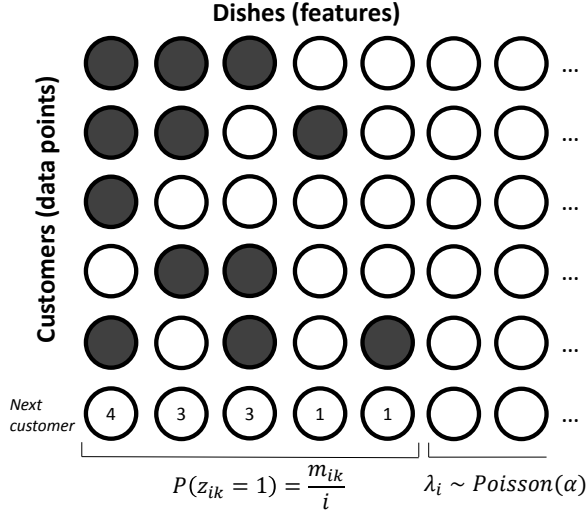


Figure 1: **Schematic of the IBP.** An example of a latent feature matrix (\mathbf{Z}) generated by the IBP. Rows correspond to customers (data points) and columns correspond to dishes (features). Gray shading indicates that a feature is active for a given data point. The last row illustrates the assignment process for a new customer; the counts for each feature (m_{ik}) are shown inside the circles for previously sampled features.

Though described sequentially, Griffiths and Ghahramani [16] showed that the resulting rows of the binary matrix are *exchangeable* (up to a permutation of the columns). This means that the order of the customers does not affect the probability of the resulting binary matrix. This is seen in the Beta-Bernoulli perspective, which we review in Section 3. In the next section, we develop a generalization of the IBP that relaxes this assumption.

2.2 The distance dependent Indian buffet process

Like the IBP, the dd-IBP is a distribution over binary latent feature matrices with a finite number of rows and an infinite number of columns. Each pair of customers has an associated distance, e.g., distance in time or space, or based on a covariate. Two customers that are close together in this distance will be more likely to share the same dishes (that is, features) than two customers that are far apart.

The dd-IBP can be understood in terms of the following sequential construction. First, each customer selects a Poisson-distributed number of dishes (feature columns). The dishes selected by a customer in this phase of the construction are said to be “owned” by this customer. A dish is either unowned, or is owned by exactly one customer. This step is akin to the selection of new dishes in the IBP.

Then, for each owned dish, customers connect to one another. The probability that one customer connects to another decreases in the distance between them. Note that customers do not sample each dish, as in the IBP, but rather connect to other customers.

Finally, dish inheritance is computed: A customer inherits a dish if its owner (from the first step) is reachable in the connectivity graph for that dish. This inheritance is computed deterministically from the connections generated in the previous step.¹ The dishes that each customer samples are those that he inherits or owns. Thus, similarity of sampled dishes between nearby customers is induced via distance-dependent connection probabilities.

We now more formally describe the probabilistic generative process of the binary matrix \mathbf{Z} . First, we introduce some notation and terminology.

- Dishes (columns of \mathbf{Z}) are identified with the natural numbers $\mathbb{N} = \{1, 2, \dots\}$. The set of dishes owned by customer i is $\mathcal{K}_i \subset \mathbb{N}$. The cardinality of this set is $\lambda_i = |\mathcal{K}_i|$. These sets are disjoint, so $\mathcal{K}_i \cap \mathcal{K}_j = \emptyset$ for $i \neq j$. The total number of owned dishes is $K = \sum_{i=1}^N \lambda_i$. The set of dishes owned by customers excluding i is $\mathcal{K}_{-i} = \cup_{j \neq i} \mathcal{K}_j$.
- Each dish is associated with a set of customer-to-customer assignments, specified by the $N \times K$ *connectivity matrix* \mathbf{C} , where $c_{ik} = j$ indicates that customer i connects to customer j for dish k . Given \mathbf{C} , the customers form a set of (possibly cyclic) directed graphs, one for each dish. The *ownership vector* is \mathbf{c}^* , where $c_k^* \in \{1, \dots, N\}$ indicates the customer who owns dish k , so $c_k^* = i \iff k \in \mathcal{K}_i$.
- The $N \times N$ distance matrix between customers is \mathbf{D} , where the distance between customers i and j is d_{ij} . A customer's self-distance is 0: $d_{ii} = 0$. We call the distance matrix *sequential* when $d_{ij} = \infty$ for $j > i$. In this special case, customers can only connect to previous customers.
- The *decay function* $f : \mathbb{R} \mapsto [0, 1]$ maps distance to a quantity, which we call *proximity*, that controls the probabilities of customer links. We require that $f(0) = 1$ and $f(\infty) = 0$. We obtain the *normalized proximity matrix* \mathbf{A} by applying the decay function to each customer pair and normalizing by customer. That is, $a_{ij} = f(d_{ij})/h_i$, where $h_i = \sum_{j=1}^N f(d_{ij})$.

Using this notation, we generate the feature indicator matrix \mathbf{Z} as follows:

1. **Assign dish ownership.** For each customer i , allocate $\lambda_i \sim \text{Poisson}(\alpha/h_i)$ unowned dishes to the customer's set of owned dishes, \mathcal{K}_i . For each $k \in \mathcal{K}_i$, set the ownership $c_k^* = i$.
2. **Assign customer connections.** For each customer i and dish $k \in \{1, \dots, K\}$, draw a customer assignment according to $P(c_{ik} = j | \mathbf{D}, f) = a_{ij}, j = 1, \dots, N$. Note that customers can connect to themselves. In this case, they do not inherit a dish unless they own it (see the next step).
3. **Compute dish inheritance.** We say that customer j *inherits* dish k if there exists a path along the directed graph for dish k from customer j to the dish's owner c_k^* (i.e., if c_k^* is reachable from j), where the directed graph is defined by column k of \mathbf{C} . The owner of a dish

¹If one insists upon a complete gastronomical metaphor, customer connectivity can be thought of as "I'll have what he's having."

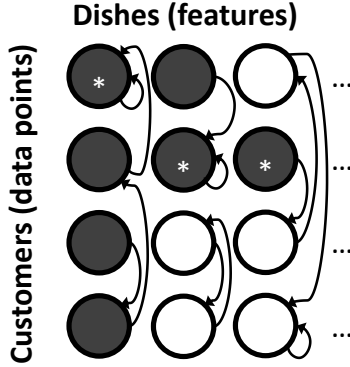


Figure 2: **Schematic of the dd-IBP.** An example of a latent feature matrix generated by the dd-IBP. Rows correspond to customers (data points) and columns correspond to dishes (features). Customers connect to each other, as indicated by arrows. Customers inherit a dish if the owner of that dish (c_k^* , indicated by stars) is reachable by a sequence of connections. Gray shading indicates that a feature is active for a given data point.

automatically inherits it.² We encode reachability with \mathcal{L} . If customer j is reachable from customer i for dish k then $\mathcal{L}_{ijk} = 1$. Otherwise $\mathcal{L}_{ijk} = 0$.

4. **Compute the feature indicator matrix.** For each customer i and dish k we set $z_{ik} = 1$ if i inherits k , otherwise $z_{ik} = 0$.

An example of customer assignments sampled from the dd-IBP is shown in Figure 2. In this example, customer 1 owns dish 1; customers 2-4 all reach customer 1, either directly or through a chain, and thereby inherit the dish (indicated by gray shading). Consequently, feature 1 is active for customers 1-4. Dish 2 is owned by customer 2; only customer 1 reaches customer 2, and hence feature 2 is active for customers 1 and 2. Dish 3 is owned by customer 2, but no other customers reach customer 2, and hence feature 3 is active only for that customer.

The generative process of the dd-IBP defines the following joint distribution of the ownership vector and connectivity matrix,

$$P(\mathbf{C}, \mathbf{c}^* | \mathbf{D}, \alpha, f) = P(\mathbf{c}^* | \alpha)P(\mathbf{C} | \mathbf{c}^*, \mathbf{D}, f). \quad (2)$$

Consider the first term. Recall that the set of dishes each customer owns \mathcal{K}_i and the total number of owned dishes K are both functions of the ownership vector \mathbf{c}^* . Thus, the probability of the ownership vector is

$$P(\mathbf{c}^* | \alpha) = \prod_{i=1}^N P(\lambda_i | \alpha), \quad (3)$$

where \mathbf{c}^* is a deterministic function of $\lambda_1, \dots, \lambda_N$.

²Although customer i can link to other customers for dish k even if $k \in \mathcal{K}_i$, these connections are ignored in determining dish inheritance when $k \in \mathcal{K}_i$.

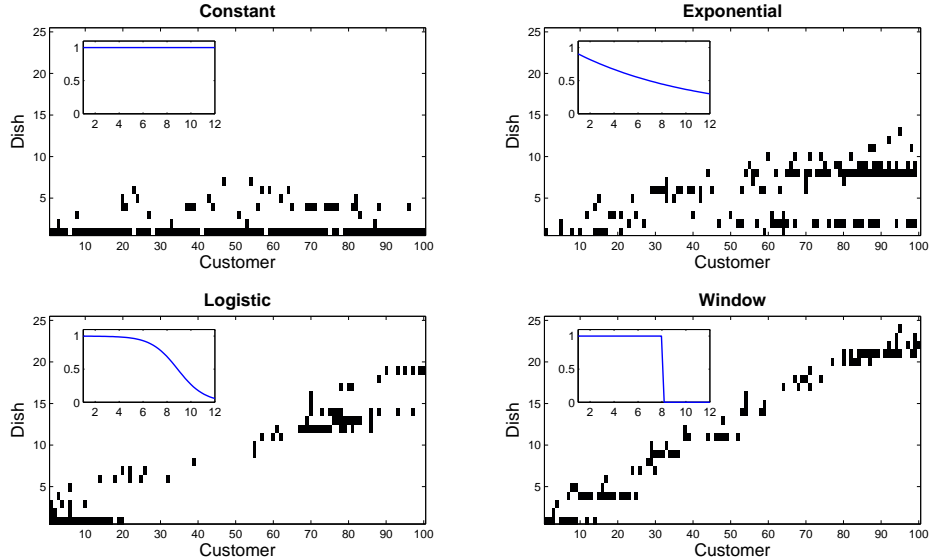


Figure 3: **Decay functions.** Each panel presents a different latent feature matrix, sampled from the dd-IBP with sequential distances. Decay functions are shown in the insets.

Consider the second term. The conditional distribution of the connectivity matrix \mathbf{C} depends on the total number of owned dishes and the normalized proximity matrix \mathbf{A} (derived from the distances and decay function),

$$P(\mathbf{C}|\mathbf{c}^*, \mathbf{D}, f) = \prod_{i=1}^N \prod_{k=1}^K a_{ic_{ik}}. \quad (4)$$

The dependence on \mathbf{c}^* comes from K , which is determined by \mathbf{c}^* .

Random feature models (and the traditional IBP) operate with a random binary matrix \mathbf{Z} . In the dd-IBP, \mathbf{Z} is a (deterministic) many-to-one function of the random variables \mathbf{C} and \mathbf{c}^* , which we denote by ϕ . We compute the probability of a binary matrix by marginalizing out the appropriate configurations of these variables

$$P(\mathbf{Z}|\mathbf{D}, \alpha, f) = \sum_{(\mathbf{c}^*, \mathbf{C}): \phi(\mathbf{c}^*, \mathbf{C}) = \mathbf{Z}} P(\mathbf{c}^*, \mathbf{C}|\mathbf{D}, \alpha, f). \quad (5)$$

The dd-IBP reduces to the standard IBP in the special case when $f(d) = 1$ for all $d < \infty$ and the distance matrix is sequential. (Recall: \mathbf{D} is sequential if $d_{ij} = \infty$ for $j > i$.) To see this, consider the probability that the k th dish is sampled by the i th customer (that is, $z_{ik} = 1$). This probability is the proportion of previous customers that already reach c_k^* because the probability of connecting to each customer is proportional to one. This probability is m_{ki}/i , which is the same as in the IBP. This is akin to the relationship between the dd-CRP and the traditional CRP under the same condition.

Many different decay functions are possible within this framework. Figure 3 shows samples of

\mathbf{Z} using four decay functions and a sequential distance defined by absolute temporal distance ($d_{ij} = i - j$ for $i \geq j$ and $d_{ij} = \infty$ for $j > i$).

- The *constant*, $f(d) = 1$. This is the standard IBP.
- The *exponential*, $f(d) = \exp(-\beta d)$.
- The *logistic*, $f(d) = 1/(1 + \exp(\beta d - \nu))$.
- The *window*, $f(d) = \mathbf{1}[d < \nu]$.

Each decay function encourages the sharing of features across nearby rows in a different way.

When combined with an observation model (which specifies how the latent features give rise to observed data), the dd-IBP functions as a prior over latent feature representations of a data set. In section 6, we consider a specific example of how the dd-IBP can be used to analyze data.

2.3 Marginal invariance and exchangeability

Unlike the traditional IBP, the dd-IBP is not (in general) *marginally invariant*, the property that removing a customer leaves the distribution over latent features for the remaining customers unchanged. (The dd-IBP builds on the dd-CRP, which is not marginally invariant either.) In some circumstances, marginal invariance is desirable for computational reasons. For example, the conditional distributions over missing data for models lacking marginal invariance require computing ratios of normalization constants. In contrast, marginally invariant models, due to their factorized structure, require less computation for conditional distributions over missing data. In other circumstances, such as exploratory analysis of fully observed datasets, this computational concern is less important. Beyond computational considerations, while marginal invariance may be an appropriate modeling assumption in some data sets, it may be inappropriate in others.

Also unlike the traditional IBP, the dd-IBP is not exchangeable in general. To state this formally, let π be a permutation of the integers $\{1, \dots, N\}$, and for a given $N \times \infty$ binary matrix \mathbf{Z} , let \mathbf{Z}^π be the matrix created by permuting its rows according to π . Let \mathbf{Z} be drawn from the dd-IBP with distance matrix \mathbf{D} , mass parameter α and decay function f . Then, except in certain special cases (such as when \mathbf{D} recovers the traditional IBP),

$$P(\mathbf{Z} = \mathbf{z} | \mathbf{D}, \alpha, f) \neq P(\mathbf{Z} = \mathbf{Z}^\pi | \mathbf{D}, \alpha, f).$$

Permuting the data changes its distribution, and so the dd-IBP is not exchangeable in general.

Although the dd-IBP is not exchangeable, it does have a related symmetry. Let \mathbf{D}^π be the $N \times N$ matrix \mathbf{D} with both its rows and its columns permuted according to π , and let \mathbf{Z}^π be drawn from the dd-IBP with distance matrix \mathbf{D}^π rather than \mathbf{D} . (We retain the same values for α and f .) Then, in general,

$$P(\mathbf{Z} = \mathbf{z} | \mathbf{D}, \alpha, f) = P(\mathbf{Z}^\pi = \mathbf{z}^\pi | \mathbf{D}, \alpha, f).$$

Thus, if we permute both the data and the distance matrix, probabilities remain unchanged. Permuting both the data and the distance matrix is like first relabeling the data, and then explicitly

altering the probability distribution to account for this relabeling. If the dd-IBP were exchangeable, one would not need to alter the probability distribution to account for relabeling.

3 Related work

In this section we describe related work on infinite latent feature models that capture external dependence between the data. We focus on the most closely related model, which is the *dependent hierarchical beta process* (dHBP; [31]). As a prelude to describing the dHBP, we review the connection between the IBP and the beta process.

3.1 The beta process

Recall that the IBP is exchangeable. Consequently, by de Finetti’s theorem [3], the rows of \mathbf{Z} , considered as binary vectors $\{\mathbf{z}_i\}$, are conditionally independent,

$$P(\mathbf{Z}) = \int \prod_{i=1}^N P(\mathbf{z}_i|B) dP(B). \quad (6)$$

In this marginal distribution, B is a random measure on the feature space Ω and $P(B)$ is the de Finetti mixing distribution (see [3]). Thibaux and Jordan [28] showed that the de Finetti mixing distribution underlying the IBP is the *beta process* (BP), parameterized by a positive *concentration parameter* c and a *base measure* B_0 on Ω . A draw $B \sim \text{BP}(c, B_0)$ is defined by a countably infinite collection of weighted atoms,

$$B = \sum_{k=1}^{\infty} p_k \delta_{\omega_k}, \quad (7)$$

where δ_{ω} is a probability distribution that places a single atom at $\omega \in \Omega$, and the $p_k \in [0, 1]$ are independent random variables whose distribution is described as follows. If B_0 is continuous, then the atoms and their weights are drawn from a nonhomogeneous Poisson process defined on the space $\Omega \times [0, 1]$ with rate measure

$$\nu(d\omega, dp) = cp^{-1}(1-p)^{c-1} dp B_0(d\omega). \quad (8)$$

If B_0 is discrete and of the form $B_0 = \sum_{k=1}^{\infty} q_k \delta_{\omega_k}$, $q_k \in [0, 1]$, then B has atoms at the same locations as B_0 , with $p_k \sim \text{Beta}(cq_k, c(1-q_k))$. Following Thibaux and Jordan [28], we define the *mass parameter* as $\gamma = B_0(\Omega)$. Note that B_0 is not necessarily a probability measure, and hence γ can take on non-negative values different from 1.

Conditional on a draw from the beta process, the feature representation X_i of data point i is generated by drawing from the *Bernoulli process* (BeP) with base measure B : $X_i|B \sim \text{BeP}(B)$. If B is discrete, then $X_i = \sum_{k=1}^{\infty} z_{ik} \delta_{\omega_k}$, where $z_{ik} \sim \text{Bernoulli}(p_k)$. In other words, feature k is activated with probability p_k independently for all data points. Sampling \mathbf{Z} from the compound beta-Bernoulli process is equivalent to sampling \mathbf{Z} directly from the IBP when $c = 1$ and $\gamma = \alpha$ [28].

3.2 Dependent hierarchical beta processes

The dHBP [31] builds external dependence between data points using the above BP construction. The dependencies are induced by mixing independent BP random measures, weighted by their proximities \mathbf{A} .

The dHBP is based on the following generative process,

$$\begin{aligned} X_i|B_{g_i}^* &\sim \text{BeP}(B_{g_i}^*), & g_i &\sim \text{Multinomial}(\mathbf{a}_i), \\ B_j^*|B &\sim \text{BP}(c_1, B), & B &\sim \text{BP}(c_0, B_0). \end{aligned} \tag{9}$$

This is equivalent to drawing X_i from a Bernoulli process whose base measure is a linear combination of BP random measures,

$$X_i|B_i \sim \text{BeP}(B_i), \quad B_i = \sum_{j=1}^N a_{ij} B_j^*. \tag{10}$$

Dependencies between data points are captured in the dHBP by the proximity matrix \mathbf{A} , as in the dd-IBP.³ This allows proximal data points (e.g., in time or space) to share more latent features than distant ones.

In Section 4, we compare the feature-sharing properties of the dHBP and dd-IBP. Using an asymptotic analysis, we show that the dd-IBP offers more flexibility in modeling the proportion of features shared between data points, but less flexibility in modeling uncertainty about these proportions.

3.3 Other non-exchangeable variants

Although still a nascent area of research, several other non-exchangeable priors for infinite latent feature models have been proposed. Williamson, Orbanz and Ghahramani [29] used a hierarchical Gaussian process to couple the latent features of data in a covariate-dependent manner. They named this model the *dependent Indian buffet process* (dIBP). Their framework is flexible: It can couple columns of \mathbf{Z} in addition to rows, while the dd-IBP cannot. However, this flexibility comes at a computational cost during inference: Their algorithm requires sampling an extra layer of variables.

Miller, Griffiths and Jordan [21] proposed a “phylogenetic IBP” that encodes tree-structured dependencies between data. Doshi-Velez and Ghahramani [11] proposed a “correlated IBP” that couples data points and features through a set of latent clusters. Both of these models relax exchangeability, but they do not allow dependencies to be specified directly in terms of distances between data. Furthermore, inference for these models requires more intensive computation than does the standard IBP. The MCMC algorithm presented by Miller et al. [21] for the phylogenetic IBP involves both dynamic programming and auxiliary variable sampling. Similarly, the MCMC algorithm for the correlated IBP involves sampling latent clusters in addition to latent features.

³Zhou et al. [31] formalize dependencies in an equivalent manner using a normalized kernel function defined over pairs of covariates associated with the data points.

Our model also incurs extra computational cost relative to the traditional IBP due to the computation of reachability (quadratic in the number of observations); however, it permits a richer specification of the dependency structure between observations than either the phylogenetic or the correlated IBP.

Recently, Ren et al. [26] presented a novel way of introducing dependency into latent feature models based on the beta process. Instead of defining distances between customers, each dish is associated with a latent covariate vector, and distances are defined between each customer’s (observed) covariates and the dish-specific covariates. Customers then choose dishes with probability proportional to the customer-dish proximity. This construction comes with a significant computational advantage for data sets where the time complexity is tied predominantly to the number of observations. The downside of this construction is that the MCMC algorithm used for inference must sample a separate covariate vector for each dish, which may scale poorly if the covariate dimensionality is high.

4 Characterizing feature-sharing

In this section, we compare the feature-sharing properties of the dHBP and dd-IBP. Two data points share a feature if that feature is active for both (i.e., $z_{ik} = z_{jk} = 1$ for $i \neq j$ and a given feature k). This analysis is useful for understanding the types of dependencies induced by the different models, and can help guide the choice of model and hyperparameter settings for particular data analysis problems. We consider an asymptotic regime in which the mass parameter is large (α for the dd-IBP and γ for the dHBP), which simplifies feature-sharing properties. Proofs of all propositions in this section may be found in the Appendix.

4.1 Feature-sharing in the dd-IBP

We first characterize the limiting distributional properties of feature-sharing in the dd-IBP as $\alpha \rightarrow \infty$. We drop the feature index k in the reachability indicator \mathcal{L}_{ijk} , writing it \mathcal{L}_{ij} . We do this because features (that is, columns of the binary matrix \mathbf{Z}) are exchangeable under the dd-IBP and, consequently, the distribution of the random vector $(\mathcal{L}_{ijk} : i, j = 1, \dots, n)$ is invariant across k .

Let $R_i = \sum_{k=1}^{\infty} z_{ik}$ denote the number of features held by data point i , and let $R_{ij} = \sum_{k=1}^{\infty} z_{ik}z_{jk}$ denote the number of features shared by data points i and j , where $i \neq j$.

Proposition 1. *Under the dd-IBP,*

$$R_i \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1) \right), \quad (11)$$

$$R_{ij} \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1) \right). \quad (12)$$

The probabilities $P(\mathcal{L}_{in} = 1)$ and $P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1)$ depend strongly on the distribution of the connectivity matrix \mathbf{C} , but do not depend on the ownership vector \mathbf{c}^* , since \mathcal{L} is independent of dish ownership.

We derive the limiting properties of R_i and R_{ij} from properties of the Poisson distribution. In this and following results, \xrightarrow{d} indicates convergence in distribution.

Corollary 1. *Let $i \neq j$. R_i and R_{ij} converge in distribution under the dd-IBP to the following constants as $\alpha \rightarrow \infty$:*

$$\frac{R_i}{\alpha} \xrightarrow{d} \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1), \quad (13)$$

$$\frac{R_{ij}}{\alpha} \xrightarrow{d} \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1), \quad (14)$$

$$\frac{R_{ij}}{R_i} \xrightarrow{d} \frac{\sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1)}{\sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1)}. \quad (15)$$

This corollary shows that the limiting fraction of shared features R_{ij}/R_i in the dd-IBP is a constant that may be different for each pair of data points i and j . In contrast, we show below that the same limiting fraction under the dHBP is random, and takes one of two values. These two values are fixed, and do not depend upon the data points i and j .

4.2 Feature-sharing in the dHBP

Here we characterize the limiting distributional properties of feature sharing in the dHBP as B_0 becomes infinitely concentrated (i.e., $\gamma \rightarrow \infty$, analogous to $\alpha \rightarrow \infty$). In this limit, feature-sharing is primarily attributable to dependency induced by the proximity matrix \mathbf{A} .

Proposition 2. *If B_0 is continuous, then under the dHBP,*

$$R_i | \mathbf{g}_{1:N} \sim \text{Poisson}(\gamma), \quad (16)$$

$$R_{ij} | \mathbf{g}_{1:N} \sim \begin{cases} \text{Poisson}\left(\gamma \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)}\right) & \text{if } g_i = g_j, \\ \text{Poisson}\left(\gamma \frac{1}{c_0 + 1}\right) & \text{if } g_i \neq g_j. \end{cases} \quad (17)$$

We derive the limiting properties of R_i and R_{ij} from properties of the Poisson distribution.

Corollary 2. *Let $i \neq j$. Conditional on $\mathbf{g}_{1:N}$, R_i and R_{ij} converge in distribution under the dHBP to the following constants as $\gamma \rightarrow \infty$:*

$$\frac{R_i}{\gamma} \xrightarrow{d} 1, \quad (18)$$

$$\frac{R_{ij}}{\gamma} \xrightarrow{d} \begin{cases} \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)} & \text{if } g_i = g_j, \\ \frac{1}{c_0 + 1} & \text{if } g_i \neq g_j, \end{cases} \quad (19)$$

$$\frac{R_{ij}}{R_i} \xrightarrow{d} \begin{cases} \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)} & \text{if } g_i = g_j, \\ \frac{1}{c_0 + 1} & \text{if } g_i \neq g_j. \end{cases} \quad (20)$$

Thus, the expected fraction of object i 's features shared with object j , R_{ij}/R_i , is a factor of $\frac{c_0+c_1+1}{c_1+1}$ bigger when $g_i = g_j$. As $c_0 \rightarrow \infty$, this fraction goes to ∞ . As $c_0 \rightarrow 0$, it goes to 1. We can obtain the unconditional fraction by marginalizing over g_i and g_j :

Corollary 3. *Let $i \neq j$. R_{ij}/R_i converges in distribution under the dHBP as $\gamma \rightarrow \infty$ to a random variable M_{ij} defined by*

$$M_{ij} = \begin{cases} \frac{c_0+c_1+1}{(c_0+1)(c_1+1)} & \text{with probability } P(g_i = g_j), \\ \frac{1}{c_0+1} & \text{with probability } P(g_i \neq g_j), \end{cases} \quad (21)$$

where $P(g_i = g_j) = \sum_{n=1}^N a_{in}a_{jn}$.

This corollary shows that as γ grows large, the fraction of shared features becomes one of two values (determined by c_0 and c_1), with a mixing probability determined by the dependency structure. Thus, the dHBP affords substantial flexibility in specifying the mixing probability (via \mathbf{A}), but is constrained to two possible values of the limiting fraction.

4.3 Feature-sharing in the IBP

For comparison, we briefly describe the feature-sharing properties under the traditional IBP.

Under the traditional IBP, by exchangeability, R_i and R_{ij} are equal in distribution to R_1 and R_{12} . The first customer draws a $\text{Poisson}(\alpha)$ number of dishes. The second customer then chooses whether to sample each of these dishes independently and with probability $1/2$. Thus, the number of dishes sampled by both the first and second customers is $R_{12} \sim \text{Poisson}(\alpha/2)$.

This shows that, under the traditional IBP, as $\alpha \rightarrow \infty$ with $i \neq j$,

$$\frac{R_i}{\alpha} \xrightarrow{d} 1, \quad \frac{R_{ij}}{\alpha} \xrightarrow{d} \frac{1}{2}, \quad \frac{R_{ij}}{R_i} \xrightarrow{d} \frac{1}{2}. \quad (22)$$

4.4 Discussion

Using an asymptotic analysis, the preceding theoretical results show that the dd-IBP and dHBP provide different forms of flexibility in specifying the way in which features are shared between data points. This asymptotic analysis takes the limit as the mass parameters α and γ become large. This limit is taken for theoretical tractability, and removes much of the uncertainty that is otherwise present in these models. While such limiting dd-IBP and dHBP models are not intended for practical use, their simplicity provides insight into behavior in non-asymptotic regimes.

Under the dd-IBP, Corollary 1 shows that the modeler is allowed a great deal of flexibility in specifying the proportions of features shared by data points. Given a matrix specifying the proportion

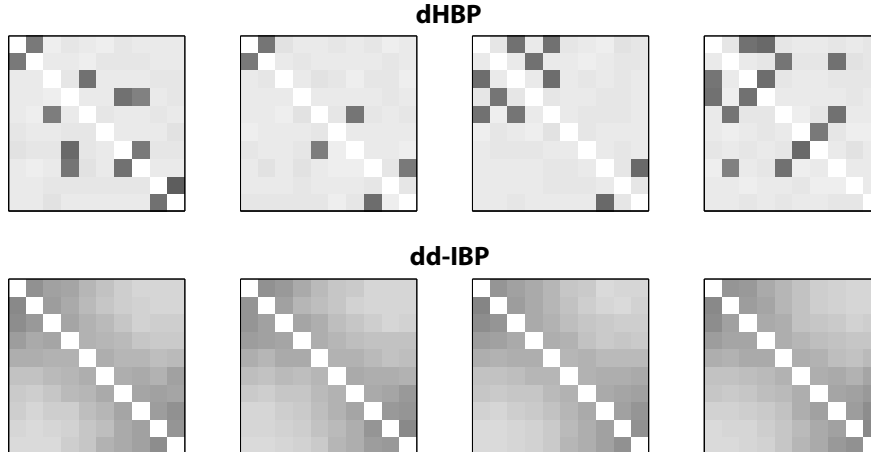


Figure 4: **Feature-sharing in the dHBP and dd-IBP, limiting case.** Along the horizontal axis, we show 4 independent draws from the dHBP (*Top*) and dd-IBP (*Bottom*). Within each subfigure, the shade of a cell (i, j) shows the fraction R_{ij}/R_i , where R_i is the number of features held by data point i , and R_{ij} is the number held by both i and j . Diagonals $R_{ii}/R_i = 1$ have been set to 0 for clarity. Here, $\alpha = \gamma = 1000$. Limiting results from Section 4 explain the behavior for such large α and γ : for the dHBP the feature-sharing proportion R_{ij}/R_i is random and equal to one of two constants; for the dd-IBP the proportion is non-random and takes a range of values. The dd-IBP models feature-sharing proportions that differ across data points, but does not model uncertainty about these proportions when mass parameters are large.

of features that are believed to be shared by pairs of data points, one can (if this matrix is sufficiently well-behaved) design a distance matrix that causes the dd-IBP to concentrate on the desired proportions. While the dd-IBP cannot model an arbitrary modeler-specified matrix of proportions, the set of matrices that can be modeled is very large.

In contrast, under the dHBP, Corollary 3 shows that the modeler has less flexibility in specifying the proportions of features shared. Under the dHBP, the modeler chooses two values, $(c_0 + c_1 + 1)/(c_0 + 1)(c_1 + 1)$ and $1/(c_0 + 1)$, and the proportion of features shared by any pair of data points in the asymptotic regime must be one of these two values.

Section 4.3 shows that the traditional IBP has the least flexibility. In the asymptotic regime, the proportion of features shared by each pair of data points is a constant.

While the dd-IBP has more flexibility in specifying values of the feature-sharing-proportions than the dHBP, it has less flexibility (at least in this asymptotic regime) in modeling uncertainty about these feature-sharing proportions. Under the dd-IBP, the proportion of features shared by a pair of data points in the asymptotic regime is a deterministic quantity. Under the dHBP, the proportion of features shared is a random quantity, even in the asymptotic regime. A modeler using the dHBP has full flexibility in choosing the joint probability distribution governing these proportions. One could extend the dd-IBP to allow uncertainty about the feature-sharing-proportions by specifying a hyperprior over distance matrices, but we do not consider this extension further.

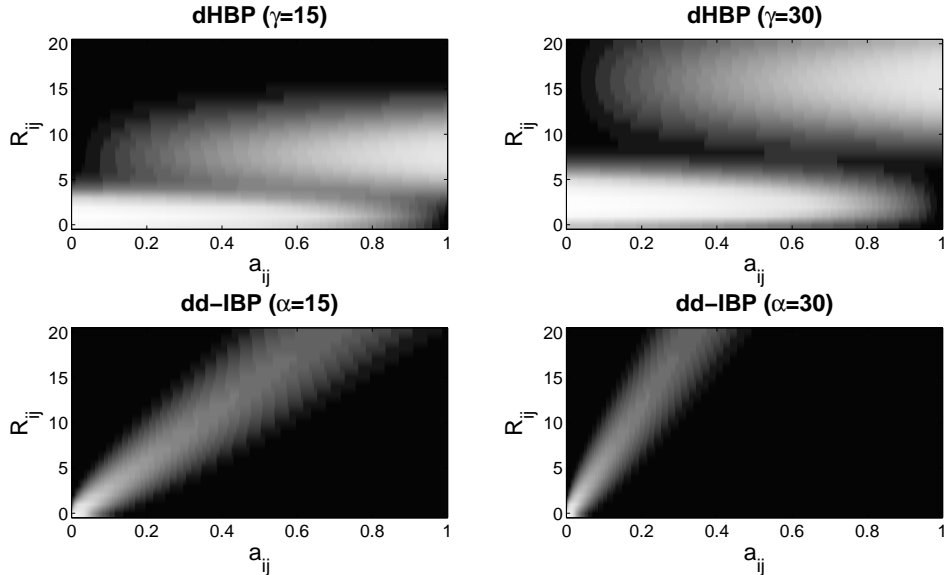


Figure 5: **Feature-sharing in the dHBP and dd-IBP.** Heatmaps of the probability mass function over the number of shared features R_{ij} (y-axis) as a function of proximity a_{ij} (x-axis) in a data set consisting of two data points. Black indicates a probability mass of 0, with lighter shades indicating larger values. For the dHBP, we set $c_0 = 10$ and $c_1 = 1$. Note that $\mathbb{E}[R_i]$ is the same for both the dHBP and dd-IBP in these examples (when $\alpha = \gamma$).

Figure 4 illustrates the difference in asymptotic feature-sharing behavior between the dHBP and dd-IBP. Subfigures in the upper row are draws from the dHBP, and subfigures in the bottom row are draws from the dd-IBP. Within a single subfigure, the shade in the cell (i, j) is the fraction R_{ij}/R_i . (The diagonals $R_{ii}/R_i = 1$ have been set to 0 to bring out other aspects of the matrix.) Each of the four columns represents a pair of independent draws. To approximate the asymptotic regime considered by the theory, the mass parameters for the two models are set to large values of $\gamma = \alpha = 1000$. The figure shows that, in draws from the dHBP, off-diagonal cells have one of two shades, corresponding to the two possible limiting values for R_{ij}/R_i . In the different columns, corresponding to different independent draws, the patterns are different, showing that R_{ij}/R_i remains random under the dHBP, even in the asymptotic regime. In contrast, in draws from the dd-IBP, off-diagonal cells take a variety of different values, but remain unchanged across independent draws.

Figure 5 illustrates non-asymptotic feature-sharing behavior in a simple setting with only two data points. The figure shows the feature-sharing behavior of the dHBP (top) and dd-IBP (bottom) at two values for the mass parameter: $\alpha = \gamma = 15$ (top row) and $\alpha = \gamma = 30$ (bottom row). Each subfigure shows the probability mass function $P(R_{ij})$ as a function of the proximity a_{ij} , where $a_{ij} = 1/d_{ij}$ for the dd-IBP. Because there are only two data points, with $a_{ii} = 1$ and $a_{ij} = a_{ji}$, specifying a_{ij} is sufficient for specifying the full proximity matrix \mathbf{A} . For the dHBP, we set $c_0 = 10$ and $c_1 = 1$. Also facilitating comparison, $\mathbb{E}[R_i]$ is the same between both models (when $\alpha = \gamma$).

Figure 5 shows that as the proximity a_{ij} increases to 1, the number of shared features R_{ij} tends to increase under both models. More precisely, $P(R_{ij})$ concentrates on larger values of R_{ij} as a_{ij}

increases. However, the way in which the probability mass functions change with a_{ij} differs between the two models. In the dd-IBP, the most likely value of R_{ij} increases smoothly, while under the dHBP it remains roughly constant and then jumps. As one varies a_{ij} across its full range, the set of most likely values for R_{ij} under the dd-IBP spans its full range from 0 to 20, while under the dHBP the most likely value for R_{ij} takes only a few values. Instead, varying a_{ij} under the dHBP allows a variety of bimodal distributions centered near the values from the asymptotic analysis.

This difference in non-asymptotic behaviors mirrors the difference between the two models in the asymptotic regime, where the dd-IBP allows feature-sharing-proportions to be specified almost arbitrarily but allows little flexibility in modeling uncertainty about them, and the dHBP limits the number of possible values for the feature-sharing proportions, but allows uncertainty over these values.

5 Inference using Markov chain Monte Carlo sampling

Given a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and a latent feature model $P(\mathbf{X}|\mathbf{Z}, \theta)$ with parameter θ , the goal of inference is to compute the joint posterior over the customer assignment matrix \mathbf{C} , the dd-IBP hyperparameter α , and likelihood parameter θ , as given by Bayes’ rule:

$$P(\mathbf{C}, \mathbf{c}^*, \theta, \alpha | \mathbf{X}, \mathbf{D}, f) \propto P(\mathbf{X} | \mathbf{C}, \mathbf{c}^*, \theta) P(\theta) P(\mathbf{C} | \mathbf{D}, f) P(\mathbf{c}^* | \alpha) P(\alpha), \quad (23)$$

where the first term is the likelihood (recall that \mathbf{Z} is a deterministic function of \mathbf{C} and \mathbf{c}^*), the second term is the prior over parameters, the third term is the dd-IBP prior over the connectivity matrix \mathbf{C} , the fourth term is the prior over the ownership vector \mathbf{c}^* , and the last term is the prior over α . In what follows, we assume that \mathbf{x}_i is conditionally independent of \mathbf{z}_j and \mathbf{x}_j for $j \neq i$ given \mathbf{z}_i and θ .

Exact inference in this model is computationally intractable. We therefore use MCMC sampling [27] to approximate the posterior with L samples. The algorithm can be adapted to different datasets by choosing an appropriate likelihood function. In the next section, we show how to adapt this algorithm to a simple linear-Gaussian model.⁴

Our algorithm combines Gibbs and Metropolis updates. For Gibbs updates, we sample a variable from its conditional distribution given the current states of all the other variables. Conjugacy allows simple Gibbs updates for θ and α . Because the dd-IBP prior is not conjugate to the likelihood, we use the Metropolis algorithm to sample \mathbf{C} and \mathbf{c}^* . We generate proposals for \mathbf{C} and \mathbf{c}^* , and then accept or reject them based on the likelihood ratio. We further divide these updates into two cases: updates for “owned” (active) dishes and updates of dish ownership.

Sampling θ . To sample the likelihood parameter θ , we draw from the following conditional distribution:

$$P(\theta | \mathbf{X}, \mathbf{C}, \mathbf{c}^*) \propto P(\mathbf{X} | \mathbf{C}, \mathbf{c}^*, \theta) P(\theta), \quad (24)$$

⁴Matlab software implementing this algorithm is available at the first author’s homepage: www.princeton.edu/~sjgershm

where the prior and likelihood are problem-specific. To obtain a closed-form expression for this conditional distribution, the prior and likelihood must be conjugate. For non-conjugate priors, one can use alternative updates, such as Metropolis-Hastings or slice sampling [27]. Generally, updates for θ will be decomposed into separate updates for each component of θ . In some cases, θ can be marginalized analytically; an example is presented in the next section.

Sampling α . To sample the hyperparameter α , we draw from the following conditional distribution:

$$P(\alpha|\mathbf{c}^*, \mathbf{D}, f) \propto P(\alpha) \prod_{i=1}^N \text{Poisson}(\lambda_i; \alpha/h_i), \quad (25)$$

where λ_i is determined by \mathbf{c}^* and the prior on α is a Gamma distribution with shape ν_α and inverse scale η_α . Using the conjugacy of the Gamma and Poisson distributions, the conditional distribution over α is given by:

$$\alpha|\mathbf{c}^*, \mathbf{D}, f \sim \text{Gamma} \left(\nu_\alpha + \sum_{i=1}^N \lambda_i, \eta_\alpha + \sum_{i=1}^N h_i^{-1} \right). \quad (26)$$

Sampling assignments for owned dishes. We update customer assignments for owned dishes (corresponding to “active” features) using Gibbs sampling. For $n = 1, \dots, N$, $i = 1, \dots, N$, and $k \in \mathcal{K}_n$, we draw a sample from the conditional distribution over c_{ik} given the current state of all the other variables:

$$P(c_{ik}|\mathbf{c}_{-i}, \mathbf{x}_i, \mathbf{c}^*, \theta, \mathbf{D}, f) \propto P(\mathbf{x}_i|\mathbf{C}, \mathbf{c}^*, \theta) P(c_{ik}|\mathbf{D}, f), \quad (27)$$

where \mathbf{x}_i is the i th row of \mathbf{X} , \mathbf{c}_i is the i th row of \mathbf{C} , and \mathbf{c}_{-i} is \mathbf{C} excluding row i .⁵ The first factor in Eq. 27 is the likelihood,⁶ and the second factor is the prior, given by $P(c_{ik} = j|\mathbf{D}, f) = a_{ij}$. In considering possible assignments of c_{ik} , one of two scenarios will occur: Either data point i reaches the owner of k (in which case feature k becomes active for i as well as for all other data points that reach i), or it does not (in which case feature k becomes inactive for i as well as for all other data points that reach i). This means we only need to consider two different likelihoods when updating c_{ik} .

Sampling dish ownership. We update dish ownership and customer assignments for newly owned dishes (corresponding to features going from inactive to active in the sampling step) using Metropolis sampling. Both a new connectivity matrix \mathbf{C}' and ownership vector $\mathbf{c}^{* \prime}$ are proposed by drawing from the prior, and then accepted or rejected according to a likelihood ratio. In more detail, the update proceeds as follows.

1. Propose $\lambda'_i \sim \text{Poisson}(\alpha/h_i)$ for each data point $i = 1, \dots, N$.
2. Set $\mathbf{C}' \leftarrow \mathbf{C}$. Then populate or depopulate it by performing, for each $i = 1, \dots, N$,

⁵We rely on several conditional independencies in this expression; for example, \mathbf{x}_i is conditionally independent of \mathbf{X}_{-i} given \mathbf{C}, \mathbf{c}^* , and θ .

⁶In calculating the likelihood, we only include the active columns of \mathbf{Z} (i.e., those for which $\sum_{n=1}^N z_{nk} > 0$).

- (a) If $\lambda'_i > \lambda_i$, insert $\lambda'_i - \lambda_i$ new dishes into \mathcal{K}_i . Then, for all $k \in [\lambda_i + 1, \lambda'_i]$ and $m \neq i$, sample c'_{mk} according to $P(c'_{mk} = j) = a_{mj}$.
- (b) If $\lambda'_i < \lambda_i$, remove $\lambda_i - \lambda'_i$ randomly selected dishes from \mathcal{K}_i .

This reallocation of dishes induces a new ownership vector \mathbf{c}^{*} .

3. Compute the acceptance ratio ζ . Because the prior (conditional on the current state of the Markov chain) is being used as the proposal distribution, the acceptance ratio reduces to a likelihood ratio (the prior and proposal terms cancel out):

$$\zeta = \min \left[1, \frac{P(\mathbf{X}|\mathbf{C}', \mathbf{c}^{*'}, \theta)}{P(\mathbf{X}|\mathbf{C}, \mathbf{c}^*, \theta)} \right]. \quad (28)$$

4. Draw $r \sim \text{Bernoulli}(\zeta)$. Set $\mathbf{C} \leftarrow \mathbf{C}'$ and $\mathbf{c}^* \leftarrow \mathbf{c}^{*'}$ if $r = 1$, otherwise leave \mathbf{C} and \mathbf{c}^* unchanged.

Iteratively applying these updates, the sampler will (after a burn-in period) draw samples from a distribution that approaches the posterior (Eq. 23) as the burn-in period grows large. The time complexity of this algorithm is dominated by the reachability computation, $O(KN^2)$, and the likelihood computation, which is $O(N^3)$ if coded naively (see [16] for a more efficient implementation using rank-one updates).

6 A linear-Gaussian model

As an example of how the dd-IBP can be used in data analysis, we incorporate it into a linear-Gaussian latent feature model (Figure 6). This model was originally studied for the IBP by Griffiths and Ghahramani [16, 17]. The observed data $\mathbf{X} \in \mathbb{R}^{N \times M}$ consist of N objects, each of which is a M -dimensional vector of real-valued object properties. We model \mathbf{X} as a linear combination of binary latent features corrupted by Gaussian noise:

$$\mathbf{X} = \mathbf{Z}\mathbf{W} + \epsilon, \quad (29)$$

where \mathbf{W} is a $K \times M$ matrix of real-valued weights, and ϵ is a $N \times M$ matrix of independent, zero-mean Gaussian noise terms with standard deviation σ_x . We place a zero-mean Gaussian prior on \mathbf{W} with covariance $\sigma_w^2 \mathbf{I}$. Intuitively, the weights capture how the latent features interact to produce the observed data. For example, if each latent feature corresponds to a person in an image, then the weight w_{km} captures the contribution of person k to pixel m .

Within the algorithm of the previous section, $\theta = \mathbf{W}$. As a consequence of our Gaussian assumptions, \mathbf{W} can be marginalized analytically, yielding the likelihood:

$$\begin{aligned} P(\mathbf{X}|\mathbf{Z}) &= \int_{\mathbf{W}} P(\mathbf{X}|\mathbf{Z}, \mathbf{W})P(\mathbf{W})d\mathbf{W} \\ &= \frac{\exp \left\{ -\frac{1}{2\sigma_x^2} \text{tr} \left(\mathbf{X}^T (\mathbf{I} - \mathbf{Z}\mathbf{H}^{-1}\mathbf{Z}^T) \mathbf{X} \right) \right\}}{(2\pi)^{NM/2} \sigma_x^{(N-K)M} \sigma_w^{KM} |\mathbf{H}|^{M/2}}, \end{aligned} \quad (30)$$

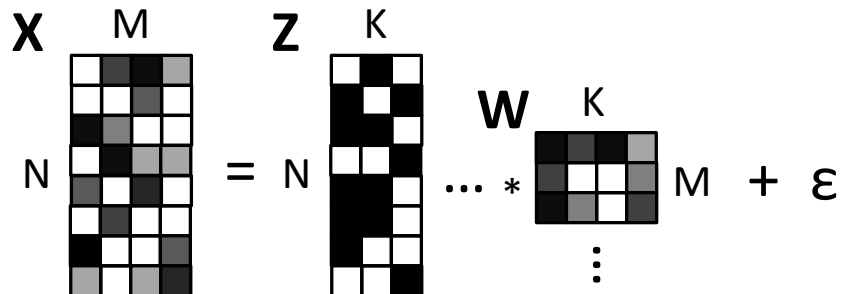


Figure 6: **Linear-Gaussian model.** Matrix multiplication view of how latent features (\mathbf{Z}) combine with a weight matrix (\mathbf{W}) and white noise (ϵ) to produce observed data (\mathbf{X}).

where $\text{tr}(\cdot)$ is the matrix trace and $\mathbf{H} = \mathbf{Z}^T \mathbf{Z} + \frac{\sigma_x^2}{\sigma_w^2} \mathbf{I}$. In calculating the likelihood, we only include the “active” columns of \mathbf{Z} (i.e., those for which $\sum_{j=1}^N z_{jk} > 0$), and K is the number of active columns.

7 Experimental results

In this section we report experimental investigations of the dd-IBP and comparisons with alternative models. We first show how the dd-IBP can be used as a dimensionality reduction pre-processing technique for classification tasks when the data points are non-exchangeable. We then show how the dd-IBP can be applied to missing data problems.

7.1 Dimensionality reduction for classification

The performance of supervised learning algorithms is often enhanced by pre-processing the data to reduce its dimensionality [4]. Classical techniques for dimensionality reduction, such as principal components analysis and factor analysis, assume exchangeability, as does the infinite latent feature model based on the IBP [16]. For this reason, these techniques may not work as well for pre-processing non-exchangeable data, and this may adversely affect their performance on supervised learning tasks.

We investigated this hypothesis using a magnetic resonance imaging (MRI) data set collected from 27 patients with Alzheimer’s disease and 35 healthy controls [8].⁷ The observed features consist of 4 structural summary statistics measured in 56 brain regions of interest: (1) surface area; (2) shape index; (3) curvedness; (4) fractal dimension. The classification task is to sort individuals into Alzheimer’s or control classes based on their observed features.

Age-related changes in brain structure produce natural declines in cognitive function that make diagnosis of Alzheimer’s disease difficult [13]. Thus, it is important to take age into account

⁷Available at: http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_July2009_ID_NI.

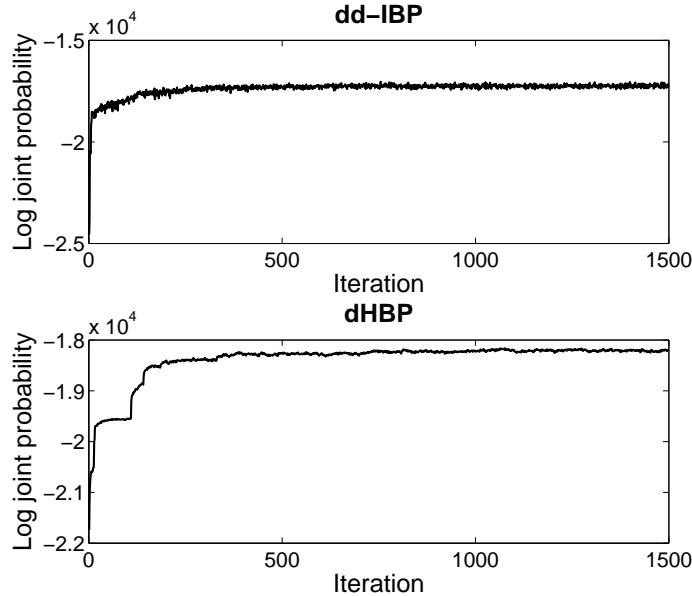


Figure 7: **Trace plots.** Representative traces of the log joint probability of the Alzheimer’s data and latent variables for the dd-IBP (*top*) and dHBP (*bottom*). Each iteration corresponds to a sweep over all the latent variables.

when designing predictive models. For the dd-IBP and dHBP, age is naturally incorporated as a covariate over which we constructed a distance matrix. Specifically, we defined d_{ij} as the absolute age difference between subjects i and j , with $d_{ij} = \infty$ for $j > i$ (i.e., the distance matrix is sequential). This induces a prior belief that individuals with similar ages tend to share more latent features. In the MRI data set, ages ranged from 60 to 90 (median: 76.5).

In detail, we ran 1500 iterations of MCMC sampling on the entire data set using the linear-Gaussian observation model, and then selected the latent features of the *maximum a posteriori* sample as input to a supervised learning algorithm (L_2 -regularized logistic regression, with the regularization constant set to 10^{-6}). Training was performed on half of the data, and testing on the other half.⁸ The noise hyperparameters of the dd-IBP and dHBP (σ_x and σ_y) were updated using Metropolis-Hastings proposals.

We monitored the log of the joint distribution $\log P(\mathbf{X}, \alpha, \mathbf{C}, \mathbf{c}^*)$. Visual inspection of the log joint probability traces suggested that the sampler reaches a local maximum within 400-500 iterations (Figure 7). This process was repeated for a range of decay parameter (β) values, using the exponential decay function. The same proximity matrix, \mathbf{A} , was used for both the dd-IBP and dHBP. We performed 5 random restarts of the sampler and recomputed the classification measure for each restart, averaging the resulting measures to reduce sampling variability. For comparison, we also made predictions using the standard IBP, the dIBP [29], and the raw observed features (i.e., no pre-processing). The dIBP was fit using the MCMC algorithm described in Williamson et al. [29], which adaptively samples the parameters controlling dependencies between observations (thus the

⁸A few randomly chosen individuals were removed from the test set to make it balanced.

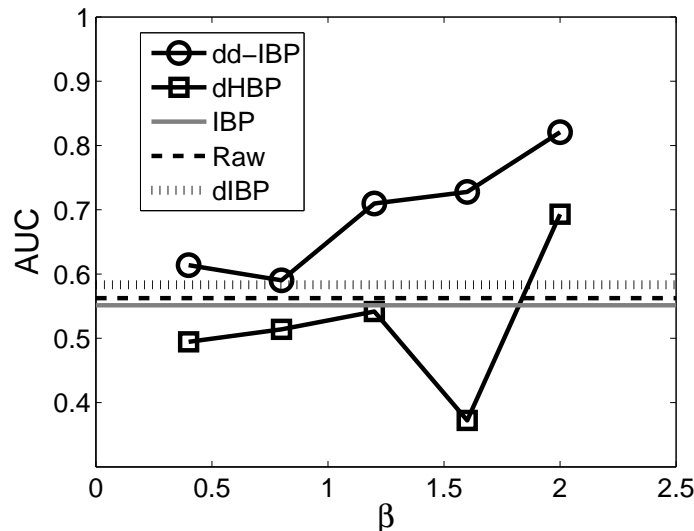


Figure 8: **Classification results for Alzheimer’s data set.** Area under the curve (AUC) for binary classification (Alzheimer’s vs. normal control) using L_2 -regularized logistic regression and features learned from a linear-Gaussian latent feature model. Each curve represents a different choice of predictor variables (latent features) for logistic regression. The x-axis corresponds to different settings of the exponential decay function parameter, β . “Raw” refers to the original data features (see text for details); the IBP, dd-IBP, dIBP and dHBP results were based on using the latent features of the *maximum a posteriori* sample following 1500 iterations of MCMC sampling.

results do not depend on β).

Classification results are shown in Figure 8 (left), where performance is measured as the area under the receiver operating characteristic curve (AUC). Chance performance corresponds to an AUC of 0.5, perfect performance to an AUC of 1. For a range of β values, the dd-IBP produces superior classification performance to the alternative models, with performance increasing as a function of β . The dHBP performs worse relative to the raw data for low β values. The magnitude of the standard error (across random restarts) is roughly 1/10 that of the means.

We also ran the dd-IBP sampler with $\beta = 0$ (in which case the dd-IBP and IBP are equivalent) and found no significant difference between it and the standard IBP sampler with respect to performance on the Alzheimer’s classification and the EEG reconstruction (see next section).

7.2 Reconstructing missing data

As an example of a missing data problem, we use latent feature models to reconstruct missing observations in electroencephalography (EEG) time series. The EEG data⁹ are from a visual detection experiment in which human subjects were asked to count how many times a particular

⁹Available at: <http://mmspl.epfl.ch/page33712.html>

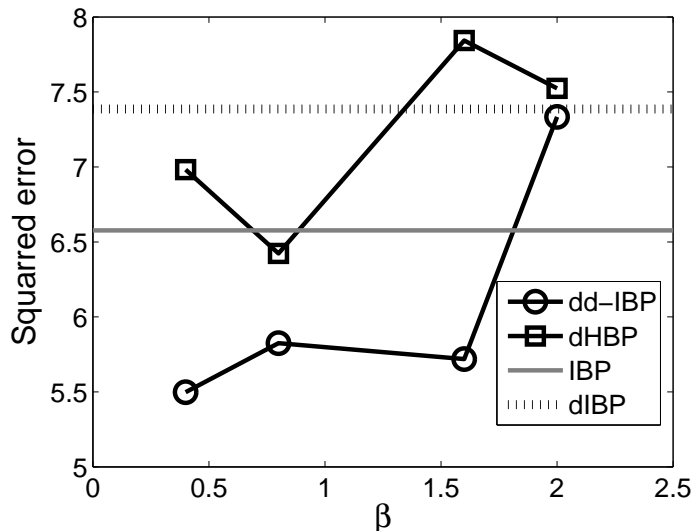


Figure 9: **Reconstruction of missing EEG data.** Reconstruction error for latent feature models as a function of the exponential decay function parameter, β . Results were based on the *maximum a posteriori* sample following 1500 iterations of MCMC sampling. Lower values indicate better performance.

image appeared on the screen [18]. The data were collected as part of a larger effort to design brain-computer interfaces to assist physically disabled subjects.

Distance between data points was defined using the absolute time-difference. Data were z-scored prior to analysis. For 10 of the data points, we removed 2 of the observed features at random. We then ran the MCMC sampler for 1500 iterations, adding Gibbs updates for the missing data by sampling from the observation distribution (Eq. 30) conditional on the current values of the latent features and hyperparameters. We then used the MAP sample for reconstruction. We measured performance by the squared reconstruction error on the missing data. Figure 9 shows the reconstruction results, demonstrating that the dd-IBP is effective for reconstructing missing data in this dataset and achieves lower reconstruction error than the alternative models we consider.

8 Conclusions

By relaxing the exchangeability assumption for infinite latent feature models, the dd-IBP extends their applicability to a richer class of data. We have shown empirically that this innovation fares better than the standard IBP on non-exchangeable data (e.g., timeseries).

We note that the dd-IBP is not a standard Bayesian nonparametric distribution, in the sense of arising from a de Finetti mixing distribution. For the standard IBP, the de Finetti mixing distribution has been identified as the beta process [28], but this result does not generalize to the dd-IBP due to its non-exchangeability (a consequence of de Finetti’s theorem). Nonetheless, this does not detract from our model’s ability to let the data infer the number of latent features, a

property that it shares with other infinite latent feature models.

A number of future directions are possible. First, it may be possible to exploit distance dependence to derive more efficient samplers. In particular, Doshi-Velez and Ghahramani [10] have shown that partitioning the data into subsets enables faster Gibbs sampling for the traditional IBP; the window decay function imposes a natural partition of the data into conditionally independent subsets.

Second, application of the dd-IBP to other likelihood functions is straightforward. For example, it could be applied to relational data [20, 22] or text data [28]. As pointed out by Miller et al. [22], covariates like age or location often play an important role in link prediction. Whereas Miller et al. [22] incorporated covariates into the likelihood function, one could instead incorporate them into the prior by defining covariate-based distances between data points (e.g., the age difference between two people). A distinction of the latter approach is that it would allow one to model dependencies in terms of latent features. For instance, two people close in age or geographic location may be more likely to share latent interests, a pattern naturally captured by the dd-IBP.

Third, modeling shared dependency structure across groups is important for several applications. In fMRI and EEG studies, for example, similar spatial and temporal dependencies are frequently observed across subjects. Modeling shared structure without sacrificing intersubject variability has been addressed with hierarchical models [2, 30]. One way to extend the dd-IBP hierarchically would be to allow the parameters of the decay function to vary across individuals while being coupled together by higher-level variables.

Acknowledgements

SJG was supported by a NSF graduate research fellowship. PIF acknowledges support from NSF Award #142251. We thank Matt Hoffman, Chong Wang, Gungor Polatkan, Sean Gerrish and John Paisley for helpful discussions. We are also grateful to Sinead Williamson and Mingyuan Zhou for sharing their code.

References

- [1] A. Ahmed and E.P. Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. In *Proc. of UAI*, 2010.
- [2] C.F. Beckmann, M. Jenkinson, and S.M. Smith. General multilevel linear modeling for group analysis in FMRI. *NeuroImage*, 20(2):1052–1063, 2003.
- [3] JM Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley & Son Ltd, 1994.
- [4] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [5] D. Blackwell and J. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.
- [6] D. Blei and P.I. Frazier. Distance dependent Chinese restaurant processes. *International Conference on Machine Learning*, 2010.

- [7] F. Caron, M. Davy, and A. Doucet. Generalized Polya urn for time-varying Dirichlet process mixtures. In *Uncertainty in Artificial Intelligence*, 2007.
- [8] N. Christou and I.D. Dinov. Confidence interval based parameter estimation a new socr applet and activity. *PLoS One*, 6(5):e19178, 2011.
- [9] Y. Chung and D.B. Dunson. The local Dirichlet process. *Annals of the Institute of Statistical Mathematics*, 63(1):59–80, 2011.
- [10] F. Doshi-Velez and Z. Ghahramani. Accelerated sampling for the Indian buffet process. In *International Conference on Machine Learning*, pages 273–280, 2009.
- [11] F. Doshi-Velez and Z. Ghahramani. Correlated non-parametric latent feature models. *Uncertainty in Artificial Intelligence*, 2009.
- [12] J.A. Duan, M. Guindani, and A.E. Gelfand. Generalized spatial Dirichlet process models. *Biometrika*, 94(4):809–825, 2007.
- [13] T. Erkinjuntti, R. Laaksonen, R. Sulkava, R. Syrjäläinen, and J. Palo. Neuropsychological differentiation between normal aging, alzheimer’s disease and vascular dementia. *Acta Neurologica Scandinavica*, 74(5):393–403, 1986.
- [14] M.D. Escobar and M. West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.
- [15] J.E. Griffin and M.F.J. Steel. Order-based dependent Dirichlet processes. *Journal of the American statistical Association*, 101(473):179–194, 2006.
- [16] T.L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. *Advances in Neural Information Processing Systems*, 18, 2005.
- [17] T.L. Griffiths and Z. Ghahramani. The Indian Buffet Process: An Introduction and Review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.
- [18] U. Hoffmann, J.M. Vesin, T. Ebrahimi, and K. Diserens. An efficient P300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, 167(1):115–125, 2008.
- [19] D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. *Independent Component Analysis and Signal Separation*, pages 381–388, 2007.
- [20] E. Meeds, Z. Ghahramani, R.M. Neal, and S.T. Roweis. Modeling dyadic data with binary latent factors. *Advances in Neural Information Processing Systems*, 19, 2007.
- [21] K.T. Miller, T.L. Griffiths, and M.I. Jordan. The phylogenetic indian buffet process: A non-exchangeable nonparametric prior for latent features. 2008.
- [22] K.T. Miller, T.L. Griffiths, and M.I. Jordan. Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*, 2009.

- [23] D.J. Navarro and T.L. Griffiths. Latent features in similarity judgments: A nonparametric Bayesian approach. *Neural computation*, 20(11):2597–2628, 2008.
- [24] V. Rao and Y.W. Teh. Spatial normalized gamma processes. *Advances in Neural Information Processing Systems*, 22:1554–1562, 2009.
- [25] C.E. Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560, 2000.
- [26] L. Ren, Y. Wang, D. Dunson, and L. Carin. The kernel beta process. *Advances in Neural Information Processing Systems*, 2011.
- [27] C.P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2004.
- [28] R. Thibaux and M.I. Jordan. Hierarchical beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, volume 11, pages 564–571. Citeseer, 2007.
- [29] S. Williamson, P. Orbanz, and Z. Ghahramani. Dependent Indian buffet processes. *International Conference on Artificial Intelligence and Statistics*, 2010.
- [30] M.W. Woolrich, T.E.J. Behrens, C.F. Beckmann, M. Jenkinson, and S.M. Smith. Multilevel linear modelling for fMRI group analysis using Bayesian inference. *NeuroImage*, 21(4):1732–1747, 2004.
- [31] M. Zhou, H. Yang, G. Sapiro, D. Dunson, and L. Carin. Dependent hierarchical beta process for image interpolation and denoising. *International Conference on Artificial Intelligence and Statistics*, 2011.

Appendix: proofs

Recall that $R_i = \sum_{k=1}^{\infty} z_{ik}$ is the number of features held by data point i , and $R_{ij} = \sum_{k=1}^{\infty} z_{ik}z_{jk}$ is the number of features shared by data points i and j , where $i \neq j$.

Proposition 1

Under the *dd-IBP*,

$$R_i \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1) \right), \quad (31)$$

$$R_{ij} \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1) \right). \quad (32)$$

Proof. For each feature, there is some probability π_i that it is turned on for data point i , and some probability π_{ij} that it is shared by i and j (note that features are exchangeable in the dd-IBP). The total number of features across all data points is distributed according to $\text{Poisson}(\lambda)$, where $\lambda = \alpha \sum_{n=1}^N h_n^{-1}$. Since each feature is turned on independently, the total number of active features for a single data point i is distributed according to $R_i \sim \text{Poisson}(\lambda\pi_i)$. Similarly, the total number of features shared by data points i and j is $R_{ij} \sim \text{Poisson}(\lambda\pi_{ij})$. The activation and co-activation probabilities are given by:

$$\pi_i = \sum_{n=1}^N P(c^* = n)P(\mathcal{L}_{in} = 1) = \frac{\sum_{n=1}^N h_n^{-1}P(\mathcal{L}_{in} = 1)}{\sum_{j=1}^N h_j^{-1}}, \quad (33)$$

$$\pi_{ij} = \sum_{n=1}^N P(c^* = n)P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1) = \frac{\sum_{n=1}^N h_n^{-1}P(\mathcal{L}_{in} = 1, \mathcal{L}_{jn} = 1)}{\sum_{j=1}^N h_j^{-1}}, \quad (34)$$

where we have used that $P(c^* = n) = h_n^{-1} / \sum_{j=1}^N h_j^{-1}$. \square

Proposition 2

If B_0 is continuous, then under the dHBP,

$$R_i | \mathbf{g}_{1:N} \sim \text{Poisson}(\gamma), \quad (35)$$

$$R_{ij} | \mathbf{g}_{1:N} \sim \begin{cases} \text{Poisson}\left(\gamma \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)}\right) & \text{if } g_i = g_j, \\ \text{Poisson}\left(\gamma \frac{1}{c_0 + 1}\right) & \text{if } g_i \neq g_j. \end{cases} \quad (36)$$

Proof. We write the random measures B and B_j^* in the generative model defining the dHBP in Section 3.2 as the following mixtures over point masses.

$$B = \sum_{k=1}^{\infty} p_k \delta_{\omega_k}, \quad p_k \sim \text{Beta}(0, c_0), \quad \omega_k \sim B_0. \quad (37)$$

$$B_j^* = \sum_{k=1}^{\infty} p_{jk}^* \delta_{\omega_k}, \quad p_{jk}^* \sim \text{Beta}(c_1 p_k, c_1(1 - p_k)). \quad (38)$$

Recall that $X_i \sim \text{BeP}(B_{g_i}^*)$ where $g_i \sim \text{Multinomial}(\mathbf{a}_i)$.

Let z_{ik} be the random variable that is 1 if the Bernoulli process draw X_i has atom ω_k , and 0 if not. We have $z_{ik} \sim \text{Bernoulli}(p_{g_i k}^*)$. Because B_0 is continuous, $P(\omega_k = \omega_{k'}) = 0$ for $k \neq k'$ and the random variables R_i and R_{ij} satisfy

$$R_i = \sum_{k=1}^K z_{ik} \quad \text{and} \quad R_{ij} = \sum_{k=1}^K z_{ik} z_{jk}. \quad (39)$$

We first show that R_i is Poisson distributed with mean γ .

Let $q_i(\epsilon)$ denote the probability that X_i has atom ω_k conditioned on $p_k > \epsilon$ (this value does not depend on k). That is, $q_i(\epsilon) = P(z_{ik} = 1 | p_k > \epsilon)$.

For a given ϵ , the density of p_k conditioned on $p_k > \epsilon$ is:

$$P(p_k \in dp | p_k > \epsilon) = \frac{c_0 p^{-1} (1-p)^{c_0-1}}{\int_{\epsilon}^1 c_0 u^{-1} (1-u)^{c_0-1} du} dp, \quad p \in (\epsilon, 1). \quad (40)$$

We can use this density to calculate the success probability $q_i(\epsilon)$:

$$q_i(\epsilon) = \mathbb{E}[z_{ik} | p_k > \epsilon] = \mathbb{E}[p_{g_i k}^* | p_k > \epsilon] = \mathbb{E}[p_k | p_k > \epsilon] = \frac{\int_{\epsilon}^1 p c_0 p^{-1} (1-p)^{c_0-1} dp}{\int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp}, \quad (41)$$

where we have used the tower property of conditional expectation in the second and third equalities.

For a given $\epsilon > 0$, let N_{ϵ} denote the number of atoms in B with $p_k > \epsilon$. This number is Poisson-distributed with mean $\lambda_{\epsilon} = \gamma \int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp$.

Let $R_i(\epsilon)$ be the number of such atoms that are also in X_i . Because $R_i(\epsilon)$ is the sum of N_{ϵ} independent Bernoulli trials that each have success probability $q_i(\epsilon)$, it follows that $R_i(\epsilon) | N_{\epsilon} \sim \text{Binomial}(N_{\epsilon}, q_i(\epsilon))$ and

$$R_i(\epsilon) \sim \text{Poisson}(\lambda_{\epsilon} q_i(\epsilon)). \quad (42)$$

Because $R_i = \lim_{\epsilon \rightarrow 0} R_i(\epsilon)$, it follows that $R_i \sim \text{Poisson}(\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_i(\epsilon))$, where

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_i(\epsilon) &= \lim_{\epsilon \rightarrow 0} \left[\gamma \int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp \right] \left[\frac{\int_{\epsilon}^1 p c_0 p^{-1} (1-p)^{c_0-1} dp}{\int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp} \right] \\ &= \lim_{\epsilon \rightarrow 0} \gamma \int_{\epsilon}^1 p c_0 p^{-1} (1-p)^{c_0-1} dp = \gamma c_0 \int_0^1 (1-p)^{c_0-1} dp = \gamma, \end{aligned}$$

where we have used that $\int_0^1 (1-p)^{c_0-1} dp = \frac{1}{c_0}$. Thus $R_i \sim \text{Poisson}(\gamma)$.

We perform a similar analysis to show the distribution of R_{ij} . Let $q_{ij}(\epsilon)$ denote the probability that X_i and X_j share atom ω_k conditional on $p_k > \epsilon$, g_i and g_j . That is,

$$q_{ij}(\epsilon) = P(z_{ik} = z_{jk} = 1 | g_i, g_j, p_k > \epsilon). \quad (43)$$

Although only ϵ appears in the argument of $q_{ij}(\epsilon)$, this quantity also implicitly depends on g_i and g_j . We calculate $q_{ij}(\epsilon)$ explicitly below.

Let $R_{ij}(\epsilon)$ be the number of atoms ω_k for which $p_k > \epsilon$ and ω_k is in both X_i and X_j . We have $R_{ij}(\epsilon) | N_{\epsilon}, g_i, g_j \sim \text{Binomial}(N_{\epsilon}, q_{ij}(\epsilon))$ and

$$R_{ij}(\epsilon) | g_i, g_j \sim \text{Poisson}(\lambda_{\epsilon} q_{ij}(\epsilon)). \quad (44)$$

Because $R_{ij} = \lim_{\epsilon \rightarrow 0} R_{ij}(\epsilon)$, it follows that $R_{ij} \sim \text{Poisson}(\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon))$.

To calculate $\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon)$, we consider two cases. In each case, we first calculate $q_{ij}(\epsilon)$ and then calculate the limit, showing that it is the same as the mean of R_{ij} claimed in the statement of the proposition.

- **Case 1:** $g_i = g_j$

$$\begin{aligned}
q_{ij}(\epsilon) &= \mathbb{E}[z_{ik}z_{jk}|p_k > \epsilon, g_i, g_j] = \mathbb{E}[(p_{g_i k}^*)^2|p_k > \epsilon, g_i, g_j] \\
&= \mathbb{E}[\mathbb{E}[(p_{g_i k}^*)^2|p_k, g_i, g_j]|p_k > \epsilon, g_i, g_j] = \mathbb{E}[p_k(c_1 p_k + 1)/(c_1 + 1)|p_k > \epsilon, g_i, g_j] \\
&= \frac{\int_{\epsilon}^1 \frac{c_1 p + 1}{c_1 + 1} p c_0 p^{-1} (1-p)^{c_0-1} dp}{\int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp} = \gamma \frac{c_0}{c_1 + 1} \frac{\int_{\epsilon}^1 (c_1 p + 1)(1-p)^{c_0-1} dp}{\lambda_{\epsilon}}. \tag{45}
\end{aligned}$$

Then the limit $\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon)$ can be written

$$\begin{aligned}
\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon) &= \gamma \frac{c_0}{c_1 + 1} \int_0^1 (c_1 p + 1)(1-p)^{c_0-1} dp \\
&= \gamma \frac{c_0}{c_1 + 1} \left[c_1 \int_0^1 p(1-p)^{c_0-1} dp + \int_0^1 (1-p)^{c_0-1} dp \right] \\
&= \gamma \frac{c_0}{c_1 + 1} \left[\frac{c_1}{c_0(c_0 + 1)} + \frac{1}{c_0} \right] = \gamma \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)},
\end{aligned}$$

where we have used that $\int_0^1 (1-p)^{c_0-1} dp = \frac{1}{c_0}$ and $\int_0^1 p(1-p)^{c_0-1} dp = \frac{1}{c_0(c_0+1)}$.

- **Case 2:** $g_i \neq g_j$

$$\begin{aligned}
q_{ij}(\epsilon) &= \mathbb{E}[z_{ik}z_{jk}|p_k > \epsilon, g_i, g_j] = \mathbb{E}[\mathbb{E}[p_{g_i k}^* p_{g_j k}^*|p_k, g_i, g_j]|p_k > \epsilon, g_i, g_j] \\
&= \mathbb{E}[p_k^2|p_k > \epsilon, g_i, g_j] = \gamma \frac{\int_{\epsilon}^1 p^2 c_0 p^{-1} (1-p)^{c_0-1} dp}{\lambda_{\epsilon}}.
\end{aligned}$$

Then the limit $\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon)$ can be written

$$\lim_{\epsilon \rightarrow 0} \lambda_{\epsilon} q_{ij}(\epsilon) = \gamma c_0 \int_0^1 p(1-p)^{c_0-1} dp = \gamma \frac{1}{c_0 + 1},$$

where we have used that $\int_0^1 p(1-p)^{c_0-1} dp = \frac{1}{c_0(c_0+1)}$.

□